



UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO

PROGRAMA ACADÉMICO DE MAESTRÍA EN INGENIERÍA DE SISTEMAS
CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Implementación de DevOps para mejorar la integración y despliegue de software en el
sector de seguros

TESIS PARA OBTENER EL GRADO ACADÉMICO DE:
Maestro en Ingeniería de Sistemas con mención en Tecnologías de la Información

AUTOR:

Bach. Fernando Enrique Casas Yeren (ORCID: 0000-0002-1208-0808)

ASESOR:

Dr. Ángel Salvatierra Melgar (ORCID: 0000-0003-2817-630X)

LÍNEA DE INVESTIGACIÓN:

Sistemas de Información y Comunicaciones

LIMA – PERÚ

2020

Dedicatoria:

Hoy, más que nunca quizás, siento gravitar sobre mí, una profunda obligación de profesional y de hombre:

¡la de adquirir nuevos conocimientos y ser libre!, si no he de ser libre hoy, no lo seré jamás. Me doy en la forma más libre que puedo en este trabajo.

Dedico este trabajo a Dios, a mi familia y a mis docentes por su apoyo incondicional durante este largo recorrido en busca de la libertad.

Agradecimientos:

Debo agradecer de manera inconmensurable a Dios, a mi familia y a mis docentes por su apoyo incondicional durante este largo recorrido.

Página del Jurado



DICTAMEN DE LA SUSTENTACIÓN DE TESIS

EL / LA BACHILLER (ES): **FERNANDO ENRIQUE CASAS YEREN**

Para obtener el Grado Académico de *Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información*, ha sustentado la tesis titulada:

IMPLEMENTACIÓN DE DEVOPS PARA MEJORAR LA INTEGRACIÓN Y DESPLIEGUE DE SOFTWARE EN EL SECTOR DE SEGUROS

Fecha: 19 de enero de 2020

Hora: 9:30 a.m.

JURADOS:

PRESIDENTE: Dr. Yolvi Ocaña Fernandez

Firma:

SECRETARIO: Dra. Liz Maribel Robladillo Bravo

Firma:

VOCAL: Dr. Angel Salvatierra Melgar

Firma:

El Jurado evaluador emitió el dictamen de:

..... **APROBADO POR UNANIMIDAD**

Habiendo encontrado las siguientes observaciones en la defensa de la tesis:

.....
.....
.....
.....

Recomendaciones sobre el documento de la tesis:

.....
.....
.....
.....

Nota: El tesista tiene un plazo máximo de seis meses, contabilizados desde el día siguiente a la sustentación, para presentar la tesis habiendo incorporado las recomendaciones formuladas por el jurado evaluador.

Declaratoria de autenticidad

Yo Fernando Enrique Casas Yeren estudiante de la Escuela de Posgrado, del programa Maestría en Gestión Pública, de la Universidad César Vallejo, Sede Lima Norte; presento mi trabajo académico titulado: “IMPLEMENTACIÓN DE DEVOPS PARA MEJORAR EL PROCESO DE INTEGRACIÓN Y DESPLIEGUE DE SOFTWARE EN EL SECTOR DE SEGUROS”, en 65 folios para la obtención del grado académico de Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información, es de mi autoría.

Por tanto, declaro lo siguiente:

- He mencionado todas las fuentes empleadas en el presente trabajo de investigación, identificando correctamente toda cita textual o de paráfrasis proveniente de otras fuentes, de acuerdo con lo establecido por las normas de elaboración de trabajos académicos.
- No he utilizado ninguna otra fuente distinta de aquellas expresamente señaladas en este trabajo.
- Este trabajo de investigación no ha sido previamente presentado completa ni parcialmente para la obtención de otro grado académico o título profesional.
- Soy consciente de que mi trabajo puede ser revisado electrónicamente en búsqueda de plagios.
- De encontrar uso de material intelectual ajeno sin el debido reconocimiento de su fuente o autor, me someto a las sanciones que determinen el procedimiento disciplinario.

Lima, 19 de enero de 2020



Fernando Enrique Casas Yeren
DNI 40973608

Índice

	Pág.
Dedicatoria	ii
Agradecimiento	iii
Página del jurado	iv
Declaratoria de autenticidad	v
Índice	vi
Índice de tablas	vii
Índice de figuras	vii
Resumen	viii
Abstract	ix
I. Introducción	1
II. Método	13
2.1. Tipo y diseño de investigación	13
2.2. Operacionalización	15
2.3. Población, muestra y muestreo	18
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad	18
2.5. Procedimiento	19
2.6. Métodos de análisis de datos	19
2.7. Aspectos éticos	19
III. Resultados	20
IV. Discusión	30
V. Conclusiones	33
VI. Recomendaciones	34
Referencias	35
Anexos	39

Índice de tablas

	Pág.
Tabla 1: Operacionalización de Variables	15
Tabla 2: Especificaciones y actividades de la variable independiente	16
Tabla 3: Validación de instrumento de medición mediante juicio de expertos	18
Tabla 4: Estadística descriptiva de los indicadores	20
Tabla 5: Prueba de normalidad de los indicadores	23
Tabla 6: Resumen de la prueba de hipótesis de las dimensiones velocidad y calidad del proceso de integración y despliegue de software	25
Tabla 7: Prueba de U Mann – Whitney para el indicador Cycletime	28
Tabla 8: Prueba de U Mann – Whitney para el indicador Frecuencia de liberación de código	28
Tabla 9: Prueba de T de Student para el indicador Ratio de éxito	29

Índice de figuras

	Pág.
Figura 1: Diagrama de cajas y bigotes del indicador Cycletime	21
Figura 2: Diagrama de cajas y bigotes del indicador Frecuencia de liberación de código	22
Figura 3: Diagrama de cajas y bigotes del indicador Ratio de éxito	22

Resumen

La presente investigación da cuenta de la implementación de DevOps en el proceso de integración y despliegue de software en una organización del sector seguros de Lima, 2019, debido a que, anterior a la implementación se tenían problemas de velocidad y calidad. El objetivo general fue determinar el efecto de DevOps sobre el proceso de integración y despliegue de software en el sector de seguros del distrito de San Isidro, Lima, 2019.

Se describe los aspectos teóricos sobre la cultura DevOps, sus prácticas y técnica tales como: integración continua, despliegue continuo, automatización de pruebas; y también de sus valores: colaboración, automatización, medición y monitoreo. Se utilizó un diseño de investigación pre-experimental y de tipo aplicada; se utilizó el método hipotético-deductivo, el enfoque cuantitativo y paradigma positivista. La población utilizada en la presente investigación es 8 sprints semanales y un total de 18 historias de usuario de un proyecto de la organización.

Para las pruebas de hipótesis se utilizó U de Man Whitney y T de Student. Teniendo como resultado que la implementación de DevOps permitió disminuir el Cycletime de 18,11 a 11,61 días en promedio, aumentar la Frecuencia de liberación de código de 1,63 a 3,25 en promedio y el aumento del Ratio de éxito de 54,58% a 72,92%. En base a los resultados se pudo concluir que se mejoró la velocidad y calidad del proceso de integración y despliegue de software en una organización del sector de seguros de San Isidro, Lima, 2019.

Palabras claves: DevOps, Integración Continua, Despliegue Continuo, Agilidad, Historias de Usuario.

Abstract

The present investigation gives account of the implementation of DevOps in the process of integration and deployment of software in an organization of the insurance sector of Lima, 2019, because, prior to the implementation there were speed and quality problems. The general objective was to identify the effect of DevOps on the Software Integration and Implementation Process in the Insurance Sector of the San Isidro district, Lima, 2019.

The theoretical aspects of the DevOps culture, its practices and techniques such as: continuous integration, continuous deployment, automation of tests; and also of its values: measurement and monitoring, automation, collaboration. A pre-experimental and applied type research design was used; the hypothetical-deductive method, the quantitative approach and positivist paradigm were used. The population used in this research is 8 weekly sprints and a total of 18 user stories of an organization project.

U of Man Whitney and T of Student were used for hypothesis tests. With the result that the implementation of DevOps allowed to reduce the Cycletime from 18,11 to 11,61 days on average, increase the Frequency of code release from 1.63 to 3.25 on average and increase the Success Ratio from 54,58% to 72,92%. Based on the results, we can conclude that the speed and quality of the software integration and deployment process in an insurance sector organization was improved, Lima, 2019.

Key words: DevOps, continuous integration, continuous deployment, Agility, User History.

I. Introducción

Puedes preguntarte a ti mismo ¿Cómo sabemos cuáles son esas capacidades que conducen al performance tecnológica y de la organización?, para (Forsgren, Humble, & Kim, 2018) la adopción de DevOps es clave, en su investigación del 2017 entre las organizaciones de alto rendimiento que adoptaron DevOps (Netflix, Amazon, Google, and Facebook) y organizaciones de bajo rendimiento se encontró una gran brecha que va creciendo continuamente, las empresas de alto rendimiento que adoptaron DevOps tienen: 46 veces más frecuencia de despliegue de código, tiempo de entrega 440 veces más rápido desde el compromiso hasta la implementación, un tiempo medio para recuperación (MTTR) 170 veces más rápido, tasa de falla de cambio 5 veces menor (1/5 de probabilidad de que falle un cambio). En Perú, el Banco BBVA en el año 2016 inicio la implementación de DevOps para acortar el ciclo de desarrollo de nuevas aplicaciones, en el 2018 ya contaba con una mejora del 45% en el tiempo de desarrollo de las nuevas aplicaciones aplicando DevOps (BBVA, 2018).

El presente estudio responde a las necesidades de aportar temática y experimentalmente la implementación de DevOps para la superar los silos existentes entre el departamentos de Desarrollo de Software (DEV) y el departamento de Infraestructura y Seguridad (OPS), específicamente en los procesos de implementación e integración de software para una empresa del sector de seguros en el distrito de San Isidro, Lima, 2019, estos silos surgen debido a que el departamento de desarrollo se encuentra centrado en la entrega software para generar valor en el negocio y el departamento de operaciones tiene como misión resguardar la seguridad y estabilidad de los activos informáticos de la empresa, este aislamiento entre las áreas mencionadas genera: reprocesos, demoras en las entregas, baja calidad, baja seguridad y poca estabilidad del software; consiste en implementar las mejores prácticas y cultura de DevOps, tales como la Integración Continua (CI) y Entrega Continua (CD); basadas en sus pilares de colaboración, automatización, lean y medición. En consecuencia, se obtiene las mejoras en el tiempo de la entrega, la calidad, la seguridad y la estabilidad del software, generando valor continuamente en la empresa.

El presente trabajo busca resolver los principales problemas del área de TI en los procesos de implementación e integración de software tales como: La demora en las

entregas de software On Premise e In Cloud son de hasta 38 y 28 días respectivamente por historia de usuario, debido a pruebas de regresión extensas y manuales que pueden durar hasta 10 días útiles cada una, procesos inadecuados para los pases entre ambientes y para la gestión de cambios, dependencia de soporte externo para los pases en todos los ambientes, los pases entre ambientes no productivos OnPremise e In Cloud pueden demorar hasta 48 horas, poca capacidad de procesamiento de los ambientes no productivos, pruebas manuales y extensas de seguridad de la información, bloqueo de los ambientes de pruebas por diferentes equipos de desarrollo.

También, baja calidad de software e inseguro por parte de los equipos debido a que los lineamientos de seguridad y arquitectura son cambiados constantemente y no son difundidos adecuadamente “teléfono malogrado”, las pruebas de vulnerabilidades se ejecutan al final del proyecto y no durante. Aplicaciones poco estables y poco escalables por un diseño no adecuado, debido a la falta de participación de infraestructura y arquitectura en cada etapa del proyecto y a la desincronización de los diferentes equipos que modifican los mismos componentes.

A través de la revisión literaria se halló estudios respecto al tema, tales como: (Carvallho Faustino, 2018), el propósito de esta investigación fue investigar cómo se puede aplicar la cultura DevOps para mejorar el proceso para gestionar los incidentes, para recolectar datos se utilizó los métodos de entrevistas semiestructuradas, análisis de documentos y observación. Llegando a la conclusión que la implementación de las prácticas de Devops tales como: pruebas automatizadas, infraestructura como código; puede ayudar a aumentar el rendimiento del equipo de Gestión de Incidentes, en la disminución severa de los tiempos de solución hasta en un 500%, mejoras en tiempos de despliegue hasta -95%, así mismo, (Nebel, 2018), cuyo propósito estudiar la aplicabilidad de la arquitectura de Microservicios para plataformas de integración, aterrizando en la conclusión que la aplicación de DevOps tiene un impacto positivo en el proceso de entrega continua de las plataformas de integración de software.

(Bejarano De la Hoz & Heredia Guerrero, 2019), tuvo como propósito de la investigación fue diseñar una estrategia de implementación e DevOps como un aporte para mejorar el software en MIPYMES de Colombia, utilizando el método descriptivo, luego a la conclusión que DevOps aporta en disminuir el ratio de Errores en las

aplicaciones y a mejorar el tiempo de desarrollo a MVP (Cycletime) en un 70%, también, (Retamal Valenzuela, 2019), el propósito de la investigación fue proveer al DCC un paradigma gestionar desarrollo y operaciones, usando técnicas DevOps, utilizando el método pre experimental. Llegando a la conclusión que el modelo propuesto en la investigación incrementa la cantidad y calidad de las aplicaciones que pasen a producción, con una disminución sustancial de proceso de entrega de software.

(Lopez Aguilar, 2015), el propósito de la investigación es presentar una solución que permita escalar rápidamente los pases a producción, a través de automatización del despliegue y de infraestructura como código, ambas técnicas usadas por DevOps, utilizando un método experimental, llegando a la conclusión que las técnicas de automatización del despliegue y de infraestructura como código, impactan favorablemente en la eficiencia de los despliegues a producción, incrementando hasta un 583% el indicador de Frecuencia de liberación de código. Así como, (Iñiguez Sánchez, 2017), el propósito de la investigación fue proponer una arquitectura que predisponga la adopción de un proceso de integración y despliegue continuo de software para la Universidad de la Cuenca, a través de un método pre experimental, llegando a las conclusiones que se debe implementar las prácticas de Test Driven Development e implementación de automatización del despliegue para mejorar calidad y tiempos de entrega del software.

(Villavicencio Zambrano, 2017), el propósito de la investigación es proponer un proceso de gestión de cambios más eficientes a través de la implementación de ITIL y Metodologías Ágiles, se utilizó un método pre experimental, llegando a la conclusión que la utilización de metodologías ágiles como scrum agregan valor y permiten cumplir con las entregas incrementales de producto minimizando riesgos, por otro lado, (Berta Hinostroza, 2011), Incorporación de Integración Continua en el Desarrollo de Software del Organismo Supervisor de la inversión en Energía y Minería, la investigación propone mejorar utilizar técnicas de integración continua para mejorar el proceso de desarrollo , utilizó el método pre experimental, llegando a la conclusión que las técnicas de integración continua influyen en el proceso de desarrollo de software.

Según GitLab, el 29% de los 5.300 desarrolladores de tecnologías de la información encuestados para el estudio 2018 ‘Global Developer Report’, planea invertir

en tecnología 'DevOps' en 2018 para acelerar sus procesos de desarrollo de 'software'. Esta tecnología, cuyo acrónimo en inglés significa 'development operations', busca crear soluciones para unificar y acelerar el proceso de desarrollo y operación de 'software', también, Además, según el portal especializado TechRepublic, el 71% de los desarrolladores que utiliza tecnología 'DevOps' califica la automatización del desarrollo de 'software' como una alta prioridad, en comparación con el 60% de los profesionales de 'Agile'. (BBVA, 2018)

El aspecto teórico se justifica en: Desde los 70's la mayoría de las empresas se encuentran en una búsqueda constante de mejores y nuevas formas de como optimizar sus procesos de desarrollo de software (Herder, Farias, & Albuquerque, 2016). En consecuencia las organizaciones han experimentado nuevas prácticas para desarrollar software rápidamente. Practicas tales como del desarrollo ágil para poderse adaptar al constante cambio de sus requerimientos (Capodiecì, Mainetti, & Manco, 2014). Así mismo, mejorar otros flujos de trabajo de la Tecnología de la Información (TI), a través de ITIL para mejorar los procesos operativos, entre otros.

Las organizaciones tradicionales frecuentemente han contado con actividades aisladas entre los departamentos de desarrollo y operaciones, cada uno de estos enfocados en sus propios objetivos; el departamento de desarrollo enfocado en entregar valor al negocio constantemente, además de corregir los errores de producción y el departamento de operaciones enfocado en mantener la disponibilidad de toda la infraestructura tecnológica, además de brindar el servicio de soporte dentro de los niveles pactados. Explicado de otra manera, los departamentos de operaciones y desarrollo tienen objetivos opuestos, mientras que desarrollo abraza el cambio, operaciones abraza la estabilidad. Esta situación es antagónica, al depender uno del otro para entregar software en producción. No debería sorprendernos que en el camino de lograr sus propios objetivos, surjan roses entre ambos departamentos, a consecuencia de la utilización de herramientas y procesos diferentes, obteniendo como resultado la desmoralización de ambos para seguir colaborando. (Hütermann, 2012).

DevOps es una cultura y no una herramienta o un flujo de trabajo predeterminado, sino una amalgama de estas, que trae consigo romper el muro que separa desarrollo de operaciones para encaminarlos al objetivo en común en base a colaboración y mejora

continua, para lograr ambos un solo objetivo, el de dar un soporte tecnológico estable, seguro y confiable para toda la organización (Davis & Daniels, 2016) y (Rajkumar, Pole, Adige, & Mahanta, 2016). Al ser DevOps percibido como una cultura, las organizaciones para contratar un Ingeniero de DevOps piensan en alguien que se especialista en muchos temas de Infraestructura, Automatización, Testing, Agilidad, Desarrollo, etc. Cuando en realidad las organizaciones primero deben preocuparse en entender de qué trata la cultura y filosofía DevOps (Kerzazi & Adams, 2016).

Devops es más complicado que entender sus beneficios, estos beneficios son claramente detallados en reportes e informes anuales como (Puppet, 2016) y (Zhu, Bass, & Champlin-Scharff, 2016). La comunidad aún no se pone de acuerdo respecto a una sola definición al respecto, de las cuales (Kerzazi & Adams, 2016) nos invita a hacernos la siguiente pregunta “¿DevOps es un rol o un equipo?” y como las instituciones educativas están formando a los futuros profesionales que tendrán este nuevo rol; así mismo, existen otros teóricos que definen a Devops como un conjunto de prácticas que tiene como objetivo reducir los tiempos de entrega entre cada flujo de desarrollo y asegurar la calidad del mismo (Virmani, 2015) . Otros teóricos, aducen que DevOps debe ser adoptada e implementada según la realidad y necesidad de cada organización y que no existe una formula estándar ni única a seguir (Nybom, Smeds, & Porres, 2016), otros lo definen como un framework (Lwakatare, Kuvaja, & Oivo, Dimensions of DevOps, 2015); sin embargo, entre todas las definiciones posibles que nos hablan de un conjunto de tecnologías y prácticas, existe una que destaca de las demás y que define a DevOps como una cultura que a través de su filosofía de trabajo redefine a una organización tradicional, hacia una organización de cultura colaborativa (Smeds, Nybom, & Porres, 2015).

DevOps es más que un framework que consiste en redefinir la cultura de los colaboradores que participan en los procesos de la organización, además de los departamentos de operaciones y desarrollo, también involucra a todos los que participen de la cadena de valor, trabajando todos enfocados en un solo objetivo para lograr los beneficios del framework. DevOps busca derribar el muro organizacional, que separa responsabilidades de tareas, y que existente entre los equipos de desarrollo y operaciones, uniéndolos con la finalidad de eliminar las trabas existentes para la velocidad de implementación de cambios y la entrega del software.

Las posibles causas del muro entre Dev(desarrollo) y Ops(operaciones) son: (1) que Dev abraza el cambio con la adopción de prácticas ágiles, sin embargo Ops está enfocado la estabilidad operativa de la infraestructura de la organización, no tienen un el objetivo en común, (2) se aplican diferentes métricas de rendimiento y desempeño entre Dev y Ops, no tienen objetivos alineados, (3) no existe una cultura de organizacional de colaboración, en consecuencia cada área debe velar por sus propios intereses y deben actuar solo dentro de sus propios límites, en lugar de adoptar la cultura de la colaboración.

DevOps tiene como objetivo principal disminuir el tiempo entre el desarrollo y cambios del software y su despliegue en producción sin descuidar la calidad, estabilidad y seguridad (Bass, Weber, & Zhu, 2015) y (Perera, Silva, & Perera, 2017), poniendo a trabajar a las áreas de desarrollo y operaciones en metas y objetivos en común, y de manera colaborativa. DevOps se propone para disminuir los problemas en común de las áreas tecnológicas, tales como: (1) el miedo natural al cambio que nace a nivel tecnológico ,así como los que nacen del proceso evolutivo del software, a causa de los riesgos que conllevan estos, (2) el tedioso proceso de implementar en producción que conlleva incertidumbre hasta que el proceso acabe, (3) el traslado de la culpa, debido al no cumplimiento del objetivo y se justifica con los bloqueos de otros equipos (Wahaballa, Abdellatief, Xiong, & Qin, 2015).

Las organizaciones tradicionales de plantean equipos de desarrollo y operaciones como silos separados, impidiendo así el escalamiento de los problemas, también caracterizándose por su resistencia a la adopción de tecnologías nuevas, con el pretexto de mantener la estabilidad, desaprovechando, inconscientemente, la oportunidad de evolucionar. Por lo contrario DevOps se caracteriza por su adopción de la innovación y la agilidad, entrando en el flujo de la mejora continua (Spinellis, 2012) y (O. White, K. B. M. Manager, and J. KabanovCEO, 2013) y (Bass L. , 2018).

DevOps está compuesto por aspectos claves como: (1) la cultura, (2) la automatización, (3) la medición, (4) la colaboración, (5) monitoreo (Nybom, Smeds, & Porres, 2016) y (Tomas, Li, & Huang, 2019). Es clave la adopción de estos aspectos claves dentro de la organización, para eso es necesario tomar acciones al respecto e identificar aquellos factores que no permitan la adopción de DevOps. (Nybom, Smeds, & Porres, 2016) Nos propone que la adopción de DevOps requiere (1) asignar responsabilidades compartidas y objetivos comunes entre las áreas de desarrollo y

operaciones, eliminando de esta manera el muro que los divide, (2) crear sinergia entre los equipos, de tal manera que unifiquen sus protocolos y refuercen la comunicación, sin desnaturalizar los roles existentes, y (3) plantear la creación de un equipo de control de DevOps que tenga por encargo conectar los equipo de desarrollo y operaciones.

La cultura: DevOps necesita una cultura basada en la colaboración, enfocada en la comunicación y la agilidad, que identifique la filosofía de cada equipo y las enfoque hacia la búsqueda de objetivos en común. La automatización: DevOps necesita se ponga foco en automatizar todos procesos posibles para poder reutilizar esos tiempos en los procesos que se requieran para lograr la generación de outcomes. La medición: DevOps necesita tomar datos en tiempo real del proceso de desarrollo de software, para realizar una medición certera del esfuerzo desplegado, para luego tener estimaciones de esfuerzo que se asemejen a la realidad, con tiempos efectivos. La colaboración: se requiere replantear la forma en que los equipos trabajan y la forma en que se comunican entre sí, en este punto es clave contar con iniciativas de agilidad para lograr una comunicación efectiva y asegurar las metas propuestas. El monitoreo: es un proceso de permanente revisión de las métricas de los procesos de desarrollo y operaciones; así como de todo el flujo, es importante implementar la obtención automática de datos de desempeño y rendimiento.

DevOps requiere de planeación continua, integración continua, despliegue continuo, pruebas continuas y un monitoreo continuo, los cuales deben realizarse en colaboración de las áreas de desarrollo y operaciones, permitiendo la información relevante para el seguimiento (Babar, Lapouchnian, & Yu, 2015). Identificado y aplicado las prácticas de DevOps de manera correcta debemos esperar que el proceso de desarrollo de software tuviera las capacidades de: (1) planeación continua, (2) despliegue continuo y colaborativo, (3) pruebas e integración continua, (4) monitoreo y optimización continuo, (5) recuperación rápida ante fallas y (6) feedback continuo del comportamiento de los usuarios (Smeds, Nybom, & Porres, 2015).

La adopción de DevOps en una organización tiene una gran dependencia de la cultura organizacional, tal como nos dice (Ebert, Gallardo, Hernantes, & Serrano, 2016). Existen evidencias características para evidenciar una óptima adopción de DevOps en una organización, entre ellas podemos mencionar: (1) objetivos y metas compartidas, (2) la

necesidad de colaboración entre equipos, (3) la innovación como estandarte, (4) Aprendizaje en base a la constante experimentación, aceptando éxitos y fallas, (5) un entorno donde los miembros se sientan parte de un gran equipo que les da soporte y apoyo constante y (6) comunicación basada la empatía y transparencia. Otros valores que deben tenerse como parte de la cultura organizacional son el respeto, la responsabilidad compartida y la confianza, (Smeds, Nybom, & Porres, 2015). Es sumamente necesario tenga claro el porqué de a adopción de DevOps, debido a que esta claridad nos sirve como punto de partida para plantear las estrategias que nos permita remover los diferentes impedimentos que puedan surgir durante esta transición, tales como (1) una organización poco o nada ágil , ni orientada a la colaboración, (2) resistencia al cambio, (3) sobrecarga laboral en los equipo de desarrollo y operaciones, (4) falta de habilidades blandas (adaptabilidad, comunicación, colaboración) (Cois, Yankel, & Connell, 2014).

El cambio cultural que DevOps facilita la identificación de metas en común entre equipos y por ende estimula la colaboración para actividades de esfuerzo en conjunto, beneficiando a la organización, en lugar del clásico modelo en que cada miembro de su equipo se encargaba de sus propias responsabilidades, asemejando sus labores como dice (Spinellis, 2012) a la entrega de sus resultados arrojándolos sobre el muro que les divide y desentendiéndose de los mismos. Para garantizar que la adopción de DevOps en su empresa no se vea afectada se debería entonces tener gran cuidado, la claridad en la definición de las razones y metas por las cuales se lleva a cabo este proceso. La estructura organizacional debe ser replanteada de manera colaborativa y por supuesto, también es necesario comprender que DevOps no siempre será la solución adecuada para todos los requerimientos de sus clientes, como podría ser el caso en que los procesos ágiles de fallo y error no son aceptables, esto y algunos aspectos profundos de los puntos críticos para procurar la salud del aspecto cultural de la organización son tratados en (Smeds, Nybom, & Porres, 2015).

El modelo de DevOps plantea aprovechar las bondades de los servicios en la nube, tales como: (1) la infraestructura como servicio (IaaS), (2) la plataforma como servicio (PaaS) y (3) el software como servicios (SaaS) para permitirse la elasticidad que la agilidad intrínseca del mismo requiere (Cois, Yankel, & Connell, 2014), lo que implica a su vez dejar a un lado el temor a las nuevas tecnologías y entender que es necesaria una transición como plantea (Woods, 2016). Una vez implementado lo planteado

anteriormente se introduce el concepto de micro servicios, que permitirán una verdadera definición arquitectural que soporte tecnológicamente DevOps (Ebert, Gallardo, Hernantes, & Serrano, 2016).

Los beneficios de la adopción de DevOps incluyen: (1) la reducción de tiempos en procesos, (2) la reutilización tanto de infraestructura como de software, (3) la escalabilidad, (4) la calidad y agilidad de entregas, entre otros beneficios (Virmani, 2015). Para permitirse adoptar DevOps, la estrategia tecnológica de la empresa debe alinearse con (1) la automatización de procesos, como fue discutido previamente en éste documento, (2) infraestructura elástica altamente escalable, para lo cual autores como (Wettinger, Breitenbucher, & Leymann, Standards-based DevOps automation and integration using TOSCA, 2015), (Lwakatare & Karvonen, Towards DevOps in the Embedded Systems Domain: Why is It So Hard?, 2016), (Christensen, 2016), (Callanan & Spillane, 2016), (Wettinger, Andrikopoulos, & Leymann, Enabling DevOps collaboration and continuous delivery using diverse application environments, 2015), (Wettinger, Breitenbücher, & Leymann, Compensation and Convergence - comparing and combining deployment automation approaches, 2015), (Wettinger, Breitenbücher, & Kopp, Streamlining DevOps automation for cloud applications using TOSCA as standardized metamodel, 2016), (Wettinger, Andrikopoulos, & Leymann, Automated capturing and systematic usage of DevOps knowledge for cloud applications, 2015).

Los autores anteriores nos proponen el uso de tecnologías en la nube, servidores virtuales y contenedores como Docker, reduciendo tiempo en procesos partiendo de la premisa de que lo que ya está hecho no hace falta volver a ser realiza; y (3) una bien definida línea base de la configuración tanto de código como infraestructura (Smeds, Nybom, & Porres, 2015) y (Zaid Abrahams & Langerman, 2018). De lo anterior que las empresas podrían ajustar las necesidades tecnológicas y de infraestructura a los valores financieros que puedan permitirse (pues se dispone de muchas herramientas de bajo presupuesto como se evidencia en (Ebert, Gallardo, Hernantes, & Serrano, 2016)) y adoptar exitosamente (desde el área tecnológica y de infraestructura) siempre que su estrategia esté bien definida y se vele por garantizar los siguientes aspectos: (1) La gestión de la configuración, (2) La gestión de la calidad en tiempo real y con mediciones aterrizadas por la retroalimentación como se habla en (Virmani, 2015), (3) La automatización de la infraestructura.

Algunos estudios muestran entre sus resultados cuáles deberían ser las habilidades ideales para los miembros que afrontarán el reto de aplicar DevOps como (Kerzazi & Adams, 2016), (Nybom, Smeds, & Porres, 2016), (Callanan & Spillane, 2016), mientras otros se enfocan desde el planteamiento de modelos para implementar DevOps según las necesidades de las organizaciones, hasta definiciones del framework (Hosono, 2012), estandarizaciones y el acercamiento de la adopción de estándares vigentes relacionados al cloud computing como lo es TOSCA (Topology and Orchestration Specification for Cloud Applications) para la estandarización de artefactos (Wettinger, Breitenbücher, & Leymann, Standards-Based DevOps Automation and Integration Using TOSCA, 2014).

A medida que DevOps es comprendido y adoptado por la organización, se debe tener especial cuidado en alinear sus metas y objetivos con las del negocio, de manera tal que se pueda percibir la verdadera naturaleza de su implementación y razón de ser en la organización. DevOps más que una entrega de software más rápido, con menos errores, con una cultura de colaboración y menos conflictos entre los desarrolladores y operaciones, es en sí, un canalizador para el entendimiento del porqué hacemos las cosas, hacia dónde va el negocio y cómo esas actividades aportan a lograr esa meta.

Según (New Relic, 2015) y (Lwakatare, Kuvaja, & Oivo, Dimensions of DevOps, 2015) DevOps cuenta con cinco motivadores críticos para la medición de su efectividad en la implementación se describen a continuación: El éxito del negocio, el software genera valor cuando se puede relacionar a un conjunto de metas del negocio, que se pueden medir, tales como, el índice de satisfacción del cliente, la tasa de conversión de potenciales clientes en clientes, ingreso promedio por cliente, costo de adquisición de clientes, entre otras. Para desarrollar o mantener software que genere valor para la organización, es crítico que esté alineado con los objetivos de negocio, de manera tal, que esta alineación se convierta en un factor de medición, que permita ayudar tanto al equipo de desarrollo, como al negocio, a entender la relación de lo que hacen mediante un conjunto de indicadores, los cuales dependen del tipo de industria en el que la empresa se encuentre, estos motivadores pueden cambiar, por ejemplo en el caso del entretenimiento y SaaS podría considerarse una métrica, las tasas de conversión, las renovaciones o las nuevas suscripciones, así como también, en algunos casos, la satisfacción del cliente a través de entregas tempranas y continuas de software de valor. La mayoría de estas métricas son del ámbito financiero y de negocio y algunas de ellas son: Retorno de la

inversión, Índice de satisfacción del cliente, Tasas de conversión, Promedio de ingresos por usuario, Costes de adquisición de clientes, Tasa de rotación de clientes, Ingresos recurrentes o incrementales, Renovaciones y/o suscripciones.

La experiencia del cliente, no basta con tener un producto innovador, alineado con el negocio y que genere valor, si funcionalmente falla, los clientes rápidamente abandonarán el producto, si es complicado, confuso, tiene muchos pasos en las transacciones importantes, puede generar frustración al usuario. Se deben llevar métricas de indicadores claves de rendimiento (KPI) para entender la experiencia del cliente y prevenir la fuga, para obtener datos para mejorar la dicha experiencia y para conocer si se está haciendo bien. Algunos indicadores serían los siguientes: Tiempos de respuesta percibidos de las transacciones clave, Frecuencia de las transacciones clave, Número de visitas por usuario/ por semana, Tasas de crecimiento de los usuarios, Tasas de conversión, Cantidad de tiempo que se pasa en la aplicación.

Estabilidad, Teniendo un software alineado con el negocio y cuya experiencia de usuario es positiva, se debe entonces medir y establecer KPI's de rendimiento, disponibilidad, escalabilidad, tiempos de respuesta, entre otros, de manera que la experiencia del usuario no se vea afectada por temas técnicos. Y no menos importante, poder responder de manera rápida ante los eventos que ocurran en producción, se debe considerar indicadores como: Uptime (disponibilidad), Tiempo de respuesta de la aplicación, Tiempo de respuesta de la base de datos, Porcentaje de tiempo de transacción en la base de datos, Utilización de los recursos, Tiempos de consulta de la base de datos.

Velocidad, La implementación con miras a ganar velocidad en el desarrollo, la entrega y el tiempo que toma corregir los errores que se presenten en producción es uno de los enfoques de DevOps, estas son algunas métricas que pueden ayudar a registrar este progreso, sin perder de vista la calidad del producto entregado y en los cuales por lo general se apoya en la Infraestructura como servicio y la nube pública, siempre teniendo en cuenta los motivadores anteriores: Tiempo de entrega de las Historias de Usuario, Frecuencia de la liberación de códigos, Tiempo medio hasta la resolución.

Calidad, Entregar de manera rápida y periódica es importante, pero más importante aún es entregar con calidad, en este punto la Integración Continua (CI), Despliegue continuo, Entrega continua y las métricas relacionadas a continuación cobran una gran importancia, sobre todo cuando se pueden medir pre y post despliegue: Tasa de éxito de despliegue, Tasa de error de aplicación, Gravedad del incidente, Errores pendientes.

Entre los problemas de la presente investigación se tiene: ¿Cuál es el efecto de DevOps sobre el Proceso de Integración y Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019?, así mismo en cuanto a los problemas específicos: (a) ¿Cuál es el efecto de DevOps sobre la velocidad de Integración y Despliegue de software?, (b) ¿Cuál es el efecto de DevOps sobre la calidad de Integración y Despliegue de software?; el presente trabajo se justifica de manera práctica que, los resultados de la presente investigación pretenden generar una mayor la adopción de DevOps en las organizaciones para mejorar en velocidad y calidad el proceso de integración y despliegue de software.

El presente trabajo se justifica de manera teórica que , según (Kim, Debois, Willis, Humble, & Allspaw, 2016) DevOps es un cambio cultural que mejora la comunicación, la colaboración y la automatización para optimizar el flujo de trabajo entre las áreas de desarrollo, operaciones de TI y el resto de la organización para asegurar entrega de extremo a extremo de los servicios digitales, teniendo como finalidad maximizar el flujo del valor para los clientes, DevOps incrementa las capacidades para diseñar, desarrollar , desplegar y operar software y servicios de forma más rápida en beneficio del negocio, logrando finalmente una alta satisfacción del cliente, una mejor calidad, una entrega más rápida y costos más bajos; también se justifica de manera práctica debido a que resultados de esta investigación pretenden evidenciar el efecto de DevOps para mejorar la velocidad y calidad del proceso de integración y despliegue de software, disminuyendo el tiempo del Cycletime e incrementando la frecuencia de liberación de código y aumentando el ratio de éxito; así mismo, a nivel de justificación metodológica, el presente trabajo permite dar nuevos alcances a otros metodólogos para realizar otras formas de medir el efecto de DevOps en el proceso de integración y despliegue de software.

Los problemas ya expuestos nos conllevan a plantear el siguiente objetivo: Identificar el efecto de DevOps sobre el Proceso de Integración y Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019, a continuación, se muestran los siguientes objetivos específicos: (a) Identificar el efecto DevOps sobre la velocidad del Proceso de Integración y Despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019, (b) Identificar el efecto de DevOps sobre la calidad del Proceso de Integración y Despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

Para contrastar nuestro estudio se plantea la siguiente hipótesis: Existe efecto de DevOps sobre el Proceso de Integración y Despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019, se muestran las hipótesis específicas: (a) Existe efecto de DevOps sobre la velocidad del Proceso de Integración y Despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019, (b) Existe efecto de DevOps sobre la calidad del Proceso de Integración y Despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019. Teniendo como objetivo general el determinar si la implementación de las practicas DevOps mejora el proceso de Integración y Despliegue de Software en empresa de Seguros.

II. Método

2.1. Tipo y diseño de la investigación

En la presente investigación se tomó en cuenta un diseño pre-experimental que según (Hernández, Fernández, & Batista, 2014) consta de una agrupación de bajo grado de control y generalmente es relevante para iniciar una aproximación al problema de investigación. Teniendo un enfoque cuantitativo, un método hipotético-deductivo y paradigma positivista, que para (Hernández, Fernández, & Batista, 2014) es una aproximación que se sostiene del razonamiento deductivo o de la lógica, iniciando con lo teórico, concibiendo a partir de estos argumentos lógicos denominados hipótesis, para que el investigador las exponga a prueba; y a través de su aplicabilidad nos ayuden a solucionar los problemas planteados. En consecuencia, para la presente tesis se aplicó un diseño pre-experimental que nos da acceso a manipular la variable independiente y estudiar como la variable dependiente afecta a las dimensiones y sus indicadores, entregándonos resultados para dar validez o no de las hipótesis planteadas, siendo el

presente trabajo de tipo aplicada, que según (Hernández, Fernández, & Batista, 2014) nos permite la utilización de teorías existentes para resolver el problema planteado.

2.2. Operacionalización de variables

Tabla 1

Operacionalización de Variables - Dimensiones e Indicadores de las variables a medir en esta investigación

Dimensión	Indicadores	Datos	Unidad de Medida	Fórmula	Niveles y Rangos	
Velocidad	Cycletime: evalúa el tiempo desde el inicio del desarrollo hasta su despliegue en producción.	Ficha de Registro	Días	CT = DD+DT+DP		
				CT= Cycletime	Bajo	3 - 11
				DD= Días de Desarrollo	Medio	12 - 20
				DT= Días de Testing	Alto	21 - 29
Calidad	Frecuencia de liberación de Código: evalúa las veces que se entre historias de usuario a producción por sprint	Ficha de Registro	Número	DP= Días de despliegue producción		
				CD=Cantidad de Despliegues por sprint	Bajo	0 - 1
					Medio	2 - 3
					Alto	4 - 5
Calidad	Ratio de Éxito: Evalúa el ratio de código desplegado de manera exitosa por cada sprint	Ficha de Registro	Porcentaje	RE = DE/TD*100%		
				TD = DE+DF	Bajo	0% - 33%
				DE=Despliegue exitosos por sprint	Medio	34% - 66%
				DF=Despliegues fallidos por sprint	Alto	67% - 100%
				TD=Total de despliegues por sprint		
				RE= Ratio de éxito		

Tabla 2

Tabla de especificaciones y actividades de la variable independiente: Implementación de Devops

Contenido Temático / Procesos	Competencia Capacidad / Resultado	Técnica de Observación	Dimensiones / Actividades	Indicadores
Identificar la Cadena de Valor Tecnológica	Cadena de Valor de ciclo de vida de Software	Observación	Reconocimiento de datos	Cicle time desde que es colocado el requerimiento hasta que se despliega en producción.
Hacer visible el trabajo	Tablero Kanban	Observación	Reconocimiento de datos	Duración del desarrollo en horas Tiempo de Pruebas en horas Tiempo de despliegue en horas
Diseñar arquitectura y organización tecnológica	Diagrama de arquitectura	Observación	Reconocimiento de datos	Tiempo de respuesta de los Micro Servicios Tiempo de respuesta de los micro fronts
Integrar Operaciones y Desarrollo	Tiempos de desarrollo y despliegue (Cycletime)	Observación	Reconocimiento de Datos	Tiempo de Cycle Time
Crear Pipeline de despliegue	Pipeline de despliegue	Observación	Reconocimiento de datos	Tiempo de despliegue automatizado al ambiente de Pruebas en minutos. Frecuencia de liberación de código por sprint.
Habilitar pruebas automatizadas rápidas y confiables	Pruebas automatizadas	Observación	Reconocimiento de datos	Tiempo de pruebas Automatizadas en el ambiente de pruebas en minutos.

Habilitar Continúa	Integración	Estrategia de Ramas en el repositorio de código fuente	Observación	Reconocimientos de datos	Cantidad de Commits por feature Cantidad de Merge por sprint.
Habilitar y automatizar entregas de bajo riesgo		Pipeline integrador con pruebas automatizadas	Observación	Reconocimientos de datos	Tiempo de despliegue Pruebas automatizadas. Ratio de éxito de despliegue.
Crear métricas para visualizar y resolver problemas		Tablero de impedimentos Tablero de Incidentes	Observación	Reconocimientos de datos	Tiempo de solución de impedimentos Tiempo de solución de incidentes
Analizar métricas para mejorar la anticipación de problemas y asegurar las metas.		Backlog de Mejoras	Observación	Reconocimientos de datos	Cantidad de historias de mejoras
Convertir los descubrimientos locales el descubrimientos globales		Ratio de Éxito	Observación	Reconocimientos de datos	Cantidad de despliegues exitosos
Incentivar las prácticas de seguridad de la información para cada colaborador.		Defectos a producción	Observación	Reconocimientos de datos	Cantidad de defectos pasados a producción.

2.3 Población

En esta investigación se utilizó una población que acorde a (Suaréz, 2011) consta de una agrupación de entes al cual se le conoce como la cuestión de estudio o al que se le quisiera dar conclusión de algún tema. Para el indicador Cycletime se tomó 8 sprint divididos en 18 historias de usuario, así mismo para los indicadores Frecuencia de Liberación de Código y Ratio de Éxito se tomó una población de 8 sprints. La población utilizada en la presente investigación es 8 sprints semanales y sus historias de usuario del proyecto Journey de Ventas Vida de una empresa de seguros de San isidro de Lima. Para la presente investigación se utilizó una muestra no probabilística de tipo censal, tomando toda la población como muestra.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

Para el presente trabajo de investigación se ha utilizado la técnica de fichaje o Instrumento de ficha de observación con la finalidad registrar la data dentro de la organización, que acorde a (Carrasco, 2006), consta de registrar data significativa y de sumo interés para la investigación, plasmando la información en tarjetas o cuadros llamados tarjetas de observación, estos datos son la resultante de la interacción directa entre el investigador y la realidad que se presenta. Se realizó la validación del instrumento de medición a través del método de juicio de expertos, el resumen de la presente evaluación se muestra en la siguiente tabla:

Tabla 3

Validación de instrumento de medición mediante juicio de expertos.

Nro.	Experto	Cycletime			Frecuencia de liberación de código			Ratio de éxito		
		Pertinencia	Relevancia	Claridad	Pertinencia	Relevancia	Claridad	Pertinencia	Relevancia	Claridad
1	Ms. Ing. Gilmar Zevallos	SI	SI	SI	SI	SI	SI	SI	SI	SI
2	MBA. Ing. Martín Bazo Rodríguez	SI	SI	SI	SI	SI	SI	SI	SI	SI
3	MBA Ing. Nelfe Dergan Febres	SI	SI	SI	SI	SI	SI	SI	SI	SI
4	Bach. Angel Nuñez Sakazar	SI	SI	SI	SI	SI	SI	SI	SI	SI
Total		SI	SI	SI	SI	SI	SI	SI	SI	SI

Fuente: Elaboración propia.

La validación se realizó presentando las fichas de observación a los expertos, teniendo como resultado la aplicabilidad y validez del instrumento para los indicadores de Cycletime, Frecuencia de liberación de código, y Ratio de éxito, los expertos seleccionados tienen amplia experiencia en el rubro de software y DevOps, siendo los expertos 1, 2 y 3 profesionales con el grado de Maestro y el experto 4 es consultor certificado de DevOps en DevOps Insitute (Organismo oficial a nivel mundial sobre temas de Devops), ver Anexo 04. Se procedió a validar la consistencia de las fichas de observación sobre la data histórica, ver Anexo 05 y Anexo 06.

2.5 Procedimiento

Para el presente trabajo se solicitó a la empresa el acceso a la información proyecto del Journey Venta Vida de desde agosto a noviembre del 2019, tomando data histórica de la base de datos de aplicaciones como: Central Agile, herramienta de gestión de proyectos ágiles de la empresa, recolectando en total 18 historias de usuario de 8 Sprint. Así como información del de la aplicación Jenkins, herramienta usada para el despliegue continuo, a través de Jobs y pipelines. También se tomó data de integración de código desde la aplicación GitLab correspondiente a los 8 Sprints. (Anexo 07)

2.6 Método de análisis de datos

Con respecto al análisis de los datos para (Hernández, Fernández, & Batista, 2014), dado a que las variables se pueden declarar en números, se hace un análisis cuantitativo empleando métodos estadísticos para el análisis de datos y de esta forma probar las hipótesis dadas. El método de análisis de datos en esta investigación es cuantitativo, ya que es pre-experimental y se obtienen estadísticas que ayuden a corroborar si la hipótesis es correcta.

2.7. Aspectos éticos

La data registrada para el presente trabajo es fidedigna a la registrada en la base de datos de la empresa. Es el compromiso del investigador del presente trabajo someterse a todos los acuerdos de confidencialidad con la organización, implementar DevOps para la mejorar el proceso de integración y despliegue de software de la empresa.

III. Resultados

Resultados descriptivos

A continuación, se muestran los resultados descriptivos de la investigación para los indicadores Cycletime, Frecuencia de liberación de código, Ratio de éxito en las siguientes tablas y figuras, en base a la data recolectada. (Anexo 05 y 06)

Tabla 4

Estadística descriptiva de los indicadores Cycletime, Frecuencia de liberación de código y Ratio de éxito.

Tabla cruzada					
Dimensión	Indicador	Niveles		Test	
				PreTest	PosTest
Velocidad	Cycletime	Bajo	% fi	28%	44%
		Medio	% fi	22%	56%
		Alto	% fi	50%	0%
			X=18,11	X=11,61	
			S=8,91	S=5,19	
	Frecuencia de despliegue de código	Bajo	% fi	50%	25%
		Medio	% fi	50%	25%
		Alto	% fi	0%	50%
			X=1,63	X=3,25	
			S=0,74	S=1,75	
Calidad	Ratio de éxito	Bajo	% fi	25%	0%
		Medio	% fi	37%	25%
		Alto	% fi	38%	75%
			X=54,58	X=72,92	
			S=34,87	S=19,47	

X = Media

S = Desviación

Fuente: Elaboración propia.

La media del indicador Cycletime en el pretest dio un valor promedio de 18,11 con una desviación de 8,91 y posterior a la implementación de DevOps para el posttest el resultado promedio fue de 11,61 con una desviación de 5,19; deduciendo en base al resultado que se redujo en 6,5. La media del indicador Frecuencia de liberación de código en el pretest dio un valor de 1,63 con una desviación de 0,74 y posterior a la implementación de DevOps para el posttest el resultado promedio fue de 3,25 con una desviación de 1,75; deduciendo en base al resultado que aumento en 1,62. La media del indicador Ratio de éxito en el pretest dio un valor promedio de 54,58 con una desviación de 34,87 y posterior a la implementación de DevOps para el posttest el resultado promedio fue de 72,92; deduciendo en base al resultado que aumento en 18,34 con una desviación de 19,47.

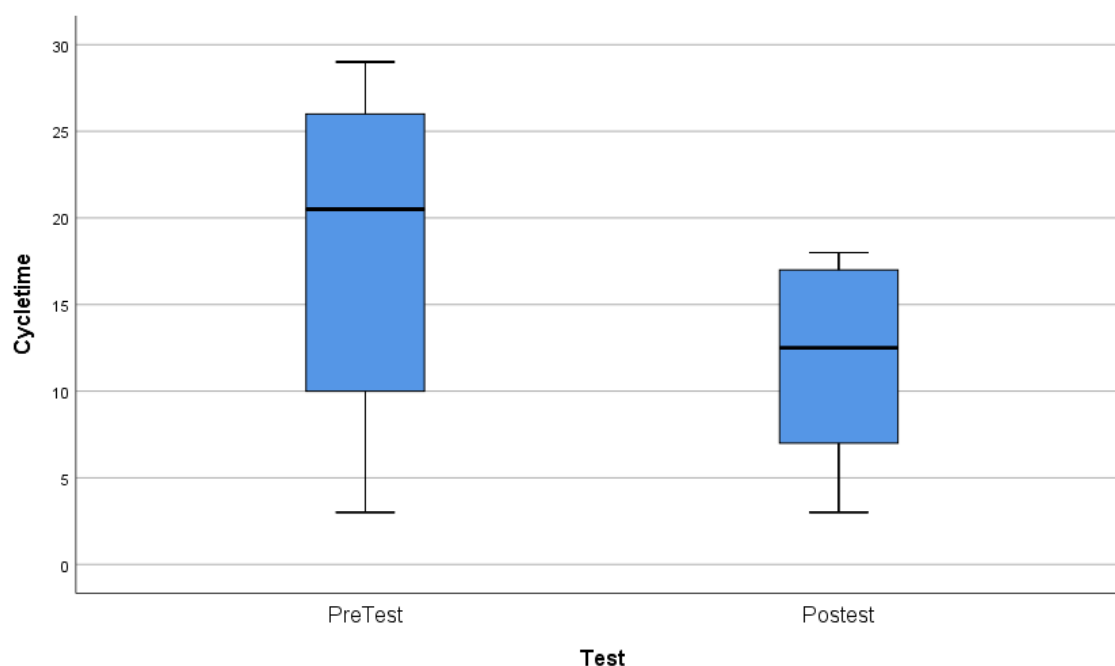


Figura 1 - Diagrama de cajas y bigotes del indicador Cycletime

En la Figura 1, se puede apreciar la comparación de los datos pretest y posttest del indicador Cycletime, evidenciando que los datos en pretest tienen mayor dispersión, caso contrario los datos en posttest tienen menor dispersión y valores menores debido a la implementación de DevOps.

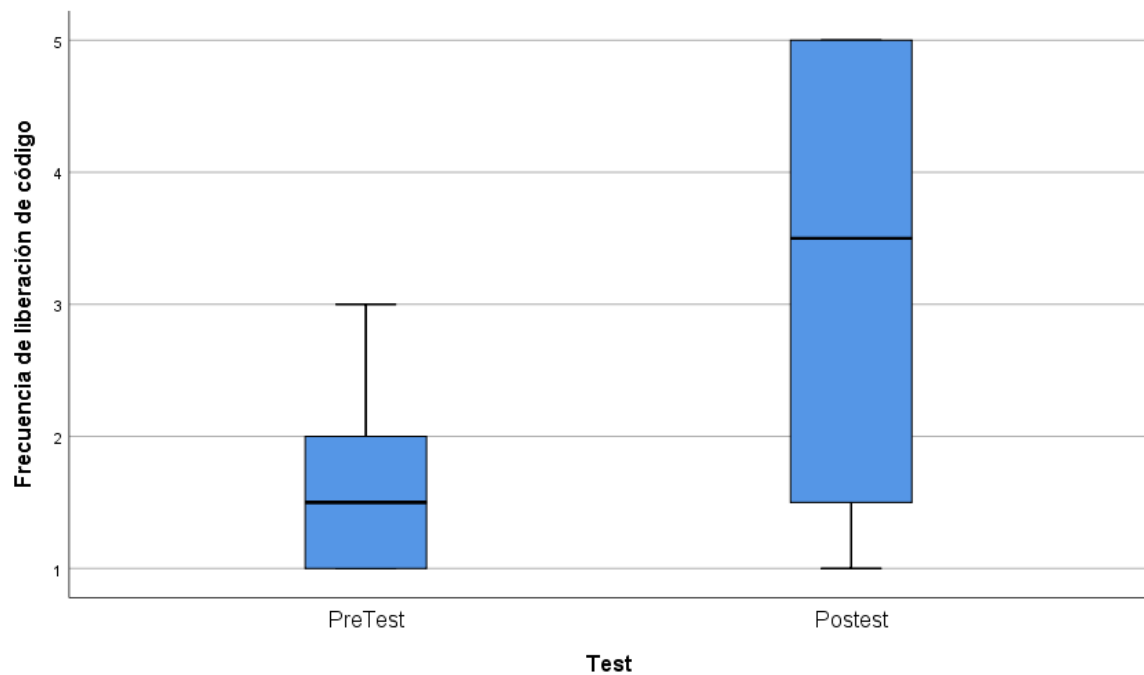


Figura 2 - Diagrama de cajas y bigotes del indicador Frecuencia de liberación de código

En la Figura 2, se puede apreciar la comparación de los datos pretest y posttest del indicador Frecuencia de liberación de código, evidenciando que los datos en pretest tienen valores menores en comparación del posttest, caso contrario los datos en posttest tienen valores mayores como efecto de la implementación de DevOps.

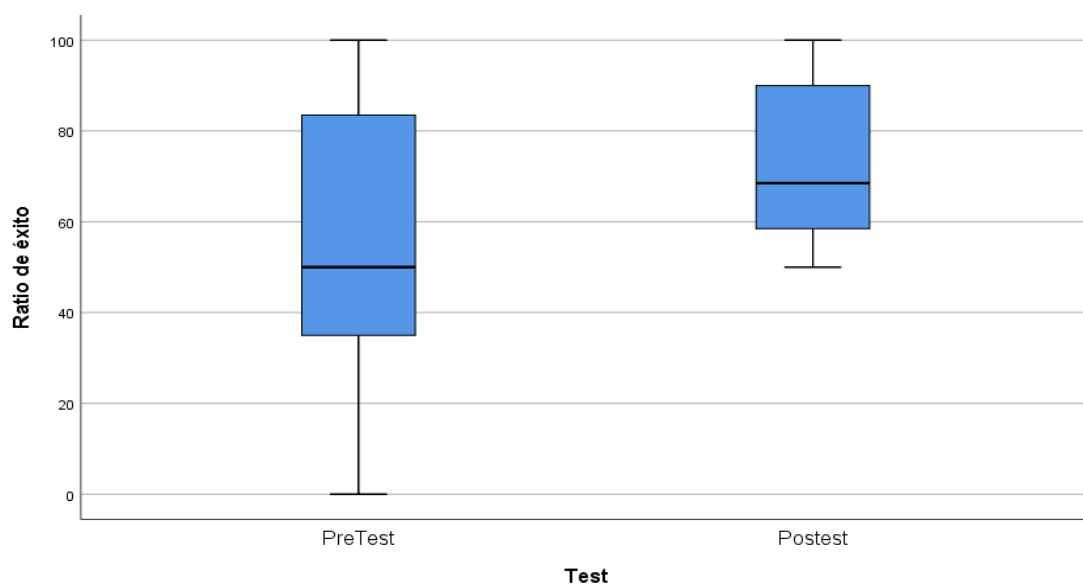


Figura 3 - Diagrama de cajas y bigotes del indicador Ratio de éxito.

De la Figura 3, se puede apreciar la comparación de los datos pretest y posttest del indicador Ratio de éxito, evidenciando que los datos en pretest tienen valores menores en comparación del posttest, caso contrario los datos en posttest tienen valores mayores como efecto de la implementación de DevOps.

Resultados inferenciales.

Análisis de los datos (Prueba de normalidad):

Se aplicó la prueba de normalidad para los datos de cada indicador, para poder determinar que prueba de hipótesis se debe realizar. Si las muestras son mayores a 30 se debe aplicar la prueba de normalidad de “Kolmogorov-Smirnov”, de ser menores o iguales a 30, se aplica la prueba de “Shapiro-Wilk”. Si el resultado de significancia es mayor o igual a 0.05 entonces la distribución de los datos es normal (paramétrica), caso contrario la distribución de los datos no es normal (no paramétrica). Entendiendo que Significancia = p – valor o nivel crítico de contraste

Debido a que la data recolectada para la presente investigación es menor a 30, se realizó la prueba de “Shapiro-Wilk”.

Tabla 5

Prueba de normalidad de los indicadores Cycletime, Frecuencia de liberación de código y Ratio de éxito.

Pruebas de normalidad				
Test		Shapiro-Wilk		
		Estadístico	gl	Sig.
Velocidad	Pretest_CycleTime	0,913	18	0,097
	Postest_CycleTime	0,890	18	0,039
	Pretest_FrecuenciaLiberacion	0,798	8	0,027
	Postest_FrecuenciaLiberacion	0,847	8	0,089
Calidad	Pretest_RatioExito	0,925	8	0,473
	Postest_RatioExito	0,890	8	0,232

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

Fuente: Elaboración propia

Los resultados en la Tabla 5 indican que el valor Sig. del indicador Cyletime para el pretest es 0.097 y en el posttest es 0.039; siendo uno de ellos menor a 0.05; en consecuencia, la distribución del indicador presenta no normalidad, por lo que asume datos no paramétricos. El resultado indica que el valor Sig. del indicador Frecuencia de liberación de código para el pretest es 0.027 y en el posttest es 0.089; siendo uno de ellos menor a 0.05; en consecuencia, la distribución del indicador presenta no normalidad, por lo que asume datos no paramétricos. El resultado indica que el valor Sig. del indicador Ratio de éxito para el pretest es 0.473 y en el posttest es 0.232; siendo ambos mayor a 0.05; en consecuencia, la distribución del indicador presenta normalidad, por lo que asume datos paramétricos.

Planteamiento de hipótesis

Los indicadores Cyletime y Frecuencia de liberación de código son de distribución no normal y son muestras independientes, se realizó la prueba de U de Mann – Whitney para la prueba de hipótesis de ambos indicadores. Así mismo, el indicador Ratio de éxito cuenta con distribución normal, en consecuencia, se le realizó la prueba de T de Student para la prueba de hipótesis.

Hipótesis general

Ho: No Existe efecto de DevOps sobre el Proceso de Integración y Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

Ho: $\mu_2 \leq \mu_1$.

Ha: Existe efecto de DevOps sobre el Proceso de Integración y Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

Ho: $\mu_2 > \mu_1$.

Tabla 6

Resumen de la prueba de hipótesis de las dimensiones velocidad y calidad del proceso de integración y despliegue de software en el sector seguros de San Isidro, Lima, 2019.

Dimensión	Indicador	Test	N	Rango promedio	Suma de rangos	Estadístico	Valor
	Cycletime	Pretest	18	18,11	32	U de Mann-Whitney	87,000
		PosTest	18	11,61	21	W de Wilcoxon	258,000
		Total	36			Sig. asintótica(bilateral)	0,017
Velocidad	Frecuencia de liberación de código	PreTest	8	1,63	4	U de Mann-Whitney	15,000
		PosTest	8	3,25	6	W de Wilcoxon	51,000
		Total	16			Sig. asintótica(bilateral)	0,046
Calidad	Ratio de Éxito	PreTest	8	54,58	100	t	-1,413
		PosTest	8	72,92	150	gl	7,000
		Total	16			Sig. (bilateral)	0,020

De la Tabla 6 podemos rechazar la hipótesis general nula y aceptar la hipótesis general alterna, debido a que la significancia del indicador Cycletime es 0,017, la significancia del indicador Frecuencia de liberación de código es de 0,046 ambos asociados a las hipótesis específica H1, así mismo, la significancia del indicador Ratio de éxito es de 0,020, indicador asociado a la hipótesis específica H2, en consecuencia se ha corroborado que la implementación de DevOps tiene efecto sobre el proceso de Integración y Despliegue de software para las organizaciones del sector seguros de San Isidro, Lima, 2019.

Hipótesis específicas

H1: Existe efecto de Devops sobre la velocidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

Indicador: Cycletime

Dónde:

CTa: Cycletime de historias de usuario antes de la implementación de DevOps.

CTd: Cycletime de historias de usuario después de la implementación de DevOps.

Hipótesis Nula (H10): No existe efecto de Devops sobre la velocidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

$$H10:CTOd - CT0a \geq 0$$

Hipótesis Alterna (H1a): existe efecto de Devops sobre la velocidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

$$H1a:CTOd - CT0a < 0$$

Indicador: Frecuencia de liberación de código

Dónde:

CDa: Frecuencia de liberación de código antes de la implementación de DevOps.

CDd: Frecuencia de liberación de código después de la implementación de DevOps.

Hipótesis Nula (H10): No existe efecto de DevOps sobre la velocidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

$$H10: CDd - CDa \leq 0$$

Hipótesis Alterna (H1a): existe efecto de DevOps sobre la velocidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

$$H1a: CDd - CDa > 0$$

H2: Existe efecto de DevOps sobre la calidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019

Indicador: Ratio de Éxito

Dónde:

REa: Ratio de Éxito antes de la implementación de DevOps.

REd: Ratio de Éxito después de la implementación de DevOps.

Hipótesis Nula (H10): No existe efecto de DevOps sobre la calidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

$$H20: REd - REa \leq 0$$

Hipótesis Alternativa (H1a): Existe efecto de DevOps sobre la calidad del proceso de integración y despliegue de software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.

$$H2a: REd - REa > 0$$

Nivel de significación de prueba

El análisis de la prueba se asume al nivel de significación estadística de $\alpha = 0.05$ con un 95% de confianza.

Selección del estadístico de prueba

Para el análisis general se asumirá la prueba de la T de Student para muestras independientes, y para los específicos se asumirá los estadísticos no paramétricos de U Maan Whitney.

Decisión:

Para asumir la decisión de la contrastación se tomó a:

$p_valor < (\alpha) = 0.05$; rechazar la hipótesis nula.

$p_valor \geq (\alpha) = 0.05$; No se rechazar la hipótesis nula.

Tabla 7

Prueba de U Mann – Whitney para el indicador Cycletime

Estadísticos de contraste ^b	
Indicador CycleTime	
U de Mann-Whitney	87,000
W de Wilcoxon	258,000
Z	-2,381
Sig. asintót. (bilateral)	0,017
Sig. exacta [2*(Sig. unilateral)]	0,017 ^a

a. No corregidos para los empates.

b. Variable de agrupación: Prueba

Fuente: Elaboración propia

Se rechaza la nula y se acepta la alterna porque el valor de Sig es menor a 0.05, con una 95% de confianza, en consecuencia, la implementación de DevOps disminuye los días de Cycletime para el proceso de integración y despliegue de software en la empresa del sector seguros del distrito de San Isidro de Lima, 2019.

Tabla 8

Prueba de U Mann – Whitney para el indicador Frecuencia de liberación de código

Estadísticos de contraste ^b	
Indicador Frecuencia de liberación de código	
U de Mann-Whitney	15,000
W de Wilcoxon	51,000
Z	-1,855
Sig. asintót. (bilateral)	0,046
Sig. exacta [2*(Sig. unilateral)]	0,083 ^a

a. No corregidos para los empates.

b. Variable de agrupación: Prueba

Fuente: Elaboración propia

Se rechaza la nula y se acepta la alterna porque el valor de Sig es menor a 0.05, con una 95% de confianza, en consecuencia, la implementación de DevOps aumenta Frecuencia de liberación de código para el proceso de integración y despliegue de software en la empresa del sector seguros del distrito de San Isidro de Lima, 2019.

Tabla 9

Prueba de T de Student para el indicador Ratio de éxito

Prueba T - Student para la igualdad de medias					
	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estandar
Ratio de éxito	-1,413	7	0,02	-18,333	12,971
Se asumen varianzas iguales					

Fuente: Elaboración propia

Se rechaza la nula y se acepta la alterna porque el valor de Sig es menor a 0.05, con una 95% de confianza, en consecuencia, la implementación de DevOps aumenta Ratio de éxito para el proceso de integración y despliegue de software en la empresa del sector seguros del distrito de San Isidro de Lima, 2019.

IV. Discusión

Como resultado de la presente investigación podemos aseverar que la implementación de DevOps mejora el proceso de integración y despliegue de software, afectando la velocidad hasta en 6,5 días menos del Cycletime, en otras palabras desde que inicia el desarrollo hasta su entrega en producción y eleva hasta en 1.62 adicional la frecuencia de liberación de código por cada sprint; también mejora la calidad, incrementando hasta un 18,34% el Ratio de éxito para el software desplegado, entendiendo por éxito que la historia de usuario desplegada en producción no requiere remediación de ningún tipo. De la misma manera (Carvallho Faustino, 2018), evidenció mejoras de tiempos de resolución de incidentes de hasta un 500% y mejoras en tiempos de despliegue 95% más rápidos, tras la implementación de la cultura DevOps,

La implementación de DevOps nos permite disminuir Cycletime el cual es el tiempo desde que se inicia el desarrollo una historia usuario, se prueba y se despliega en producción para validar una hipótesis respecto a ella misma, finalmente esto aporta gran agilidad y valor a la empresa. De la misma manera (Bejarano De la Hoz & Heredia Guerrero, 2019), DevOps como estrategia de aporte a la calidad de productos de software en MIPYMES desarrolladoras en el contexto colombiano, obtuvo una mejora del Cycletime del 70%, de esta manera las organizaciones que implementen DevOps pueden beneficiarse al entregar valor más rápido por cada sprint acelerando el ciclo de la mejora continua.

También, las técnicas de automatización del despliegue a través de pipelines utilizadas por el proyecto Journey Ventas Vida de la organización en estudio, ayudo a incrementar el indicador de Frecuencia de liberación de código de un valor de 1,63 liberaciones por sprint a 3,25 liberaciones por sprint después de la implementación de DevOps, De la misma forma (Nebel, 2018) en su tesis Arquitectura de Microservicios para Plataformas de Integración, evidencia que la implementación de DevOps influye en el tiempo de despliegue y liberación de código a través de la orquestación de despliegue de contenedores y microservicios mediante pipelines. Al igual que, (Retamal Valenzuela, 2019), en su tesis Plataformas de desarrollo de aplicaciones en el DCC basada en técnicas DevOps, pudo evidenciar que implementar estas técnicas ayudo en incrementar la cantidad de entregables y su calidad en diferentes etapas de ciclo de desarrollo de

software hasta su pase a producción.

Así mismo, en la presente investigación se evidencia la utilización de técnicas de automatización de los despliegues a diferentes ambientes, en consecuencia se la logrado incrementar la frecuencia de liberación de código al 201% en cada sprint, de la misma manera (Lopez Aguilar, 2015), en su tesis Aprovechamiento automático de infraestructura y de configuración de la infraestructura, el Caso Adblock plus, evidencia un incremento en este mismo indicador en un 583%, en consecuencia las empresas que puedan adoptar DevOps e implementan la técnica de despliegue automatizado a través de pipelines, se verán beneficiadas en el incremento de la frecuencia de liberación de código independientemente de la industria a la que pertenezcan. Y según se incorporen más prácticas de la cultura DevOps una organización tecnológica puede ser considerada de High Performance, las cuales pueden tener desempeños muy altos tales como los evidenciados en el DevOps Reports del 2018, tales como: 46 veces mayor frecuencia de liberación de código, 440 veces más rápido para el cycletime y leadtime, 170 veces más rápido del tiempo de medio de recuperación y 5 veces menos defectos a producción.

También la adopción de las prácticas de despliegue a través automatización de pruebas, y su uso a través de los pipelines de despliegue en diferentes ambientes, además de técnicas de Testing Driven Development, generan mejoras en los indicadores de las dimensiones de Tiempo y Calidad, tal como se sustentado en la presente investigación. Lo mismo demuestra (Iñiguez Sánchez, 2017) en su tesis Arquitectura tecnológica para la entrega continua de software para la Universidad de la Cuenca, Chile. La agilidad en su marco Scrum es uno de los pilares del cual se sostiene DevOps, pieza clave para poder optimizar el flujo de trabajo, destrabando las dependencias y reduciendo el Batch Size a trabajar y permitiendo minimizar riesgos y un avance hacia el cumplimiento de los objetivos del Squad y la organización, en la presente investigación se habla de Historias de usuario, Sprints lo cual evidencia el uso de Scrum para abrazar el cambio que requiere la organización. Así mismo, (Villavicencio Zambrano, 2017), en su tesis Modelo de gestión para el control de cambios y la dirección de proyectos basados en ITIL y Metodologías Ágiles de un departamento de desarrollo de Software, evidencia mejora en los indicadores de tiempo y velocidad después de la adopción metodologías ágiles(Scrum).

La presente investigación demostró el efecto de la implementación de DevOps sobre el proceso de Integración y Despliegue de software, dicha mejora se vio reflejada en la optimización de indicadores de Velocidad y Calidad, tales como la disminución del Cycletime (Tiempo desde que se inicia el desarrollo del requerimiento hasta su puesta en producción) de 18,11 a 11.61 días promedio por cada requerimiento, también el incremento del indicador Frecuencia de liberación de código (Frecuencia del código que se libera a producción por sprint) de 1,63 a 3.25 promedio de liberación de código por sprint, prácticamente duplicando el estadio inicial, así como, el incremento del Ratio de éxito (El ratio que tiene éxito un pase a producción y que no requiere un correctivo pos pase) de 18.34% al 54,58% triplicando el éxito del estadio actual, estos generosos resultados se obtuvieron tan solo en 4 meses de iniciada la implementación de DevOps, aun quedando mucho camino por recorrer en la cultura DevOps para que la organización pueda ser catalogada como una organización de alta eficiencia.

La mejora que evidencia el presente trabajo en la velocidad y la calidad del proceso de integración y despliegue de software es consecuencia de las prácticas implementadas en el proceso; tal como nos lo describe (Kim, Debois, Willis, Humble, & Allspaw, 2016), dentro de las practicas implementadas en la organización según el manual de DevOps tenemos: Mejorar el flujo de trabajo, mediante de la adopción de las practicas agiles como Scrum para evidenciar el trabajo mediante la utilización de tableros kanban, así como, asegurar la mejora continua a través del feedback constante y también, disminuir el batch size (tamaño del paquete de trabajo), asegurando un flujo constante de trabajo, permitiendo finalmente liberar código de manera fluida y mejorando los tiempos de todo el ciclo de desarrollo.

También se implementaron prácticas de Automatización para optimizar el tiempo en las tareas repetitivas y eliminar desperdicio a través de la automatización de las pruebas de regresión usando Selenium, automatización de pruebas de código estático mediante SonarQube, automatización de prueba de vulnerabilidades de secretos en código mediante el uso de Yield, y el control de las versiones del código y la gestión de ramas a través del uso del repositorio Gitlab. Finalmente, el uso de pipelines en Jenkins para orquestar todo el proceso de integración y despliegue del software mejorando considerablemente el tiempo de desarrollo, la cantidad de liberación de código y el éxito de los despliegues a producción.

Según (Forsgren, Humble, & Kim, 2018) califica a las empresas de alta eficiencia en DevOps como organización con logros e indicadores muy elevados tales como: 46 veces más frecuencia de despliegue de código, 440 veces más rápido, tiempo de entrega 440 veces más rápido desde el compromiso hasta la implementación, tiempo medio para recuperación (MTTR) 170 veces más rápido, tasa de falla de cambio 5 veces menor (1/5 de probabilidad de que falle un cambio). Estos logros son evidenciados en el reporte DevOps Reports, el cual se realiza de manera anual y comprende a la mayoría de las empresas del sector software de EEUU y Europa. También debemos tener en cuenta la los tiempos en que las empresas lograron alcanzar la categoría de alta eficiencia en DevOps, podría tomar años a la organización lograr la alta eficiencia, debido a la complejidad existente de sus procesos y su estructura organizacional, sin embargo, aquellas empresas Unicornio que adoptaron DevOps desde su nacimiento, lograron calificar como empresas de alta eficiencia en DevOps en solo 2 años en promedio, reafirmando que la implementación de DevOps tiene un gran impacto y beneficios en el proceso de integración y despliegue de software.

V. Conclusiones

En base a los resultados obtenidos en la presente investigación se concluye:

Primera:

La implementación de DevOps mejora el proceso integración y despliegue de software de la empresa de seguros de San Isidro, Lima, 2019; pues permitió aumentar la velocidad y calidad del proceso de integración y despliegue de software; y así permitió lograr los objetivos de la presente investigación y confirmar las hipótesis planteadas.

Segunda:

La implementación de DevOps disminuyó el Cycletime en 6,5 días, pues antes de la implementación (pretest) se obtuvo el valor de 18,11 días, y después (postest) se alcanzó el valor de 11,61 días; y aumentó la frecuencia de liberación de código en 1,62 por sprint, pues antes de la implementación (pretest) se obtuvo el valor de 1,63 liberaciones por sprint, y después (postest) se alcanzó el valor de 3,25 liberaciones por sprint. En consecuencia, se afirma que la implementación de DevOps mejora la velocidad para el proceso de integración y despliegue de software en el sector de seguros de San Isidro, Lima, 2019.

Tercera:

La implementación de DevOps aumentó el Ratio de éxito en 18,34%, pues antes de la implementación (pretest) se obtuvo el valor de 54,58%, y después (postest) se alcanzó el valor de 72,92%. En consecuencia, se afirma que la implementación de DevOps mejora la calidad para el proceso de integración y despliegue de software en el sector de seguros de San Isidro, Lima, 2019.

VI. Recomendaciones

Primera:

En base a los hallazgos en la presente investigación, se recomienda que, las organizaciones del sector seguros implementen DevOps, para mejorar la velocidad y la calidad en el proceso de integración y despliegue de software, a consecuencia serán más competitivos en el tiempo de entrega y la calidad de sus aplicaciones.

Segunda:

También, a consecuencia de la implementación puedan mejorar el cycletime, disminuyendo el tiempo en que se entrega una historia de usuario, disparando el pronto aprendizaje en base a la validación o no de las hipótesis sobre la historia de usuario y generando outcome y/o Quicks Wins en la empresa, permitiendo esto último tomar las decisiones correctas para lograr cumplir los OKR o KPI; así mismo, puedan incrementar la frecuencia de despliegue de código a producción, también generando rápido feedback sobre su uso por parte del usuario final para mejorar continuamente el software entregado y lanzando nuevas hipótesis por cada Sprint iniciando un espiral continuo de la experimentación y mejora.

Tercera:

También, puedan mejorar el Ratio de éxito de los despliegues, esto quiere decir que los despliegues hechos no requieran remediación, impactando esto en la estabilidad del ambiente, el Uptime y menos tiempo de correcciones en caliente, mejorando enormemente la adopción de los usuarios finales.

VII. Referencias

- Babar, Z., Lapouchnian, A., & Yu, E. (05 de 11 de 2015). Modeling DevOps Deployment Choices Using Process Architecture Design Dimensions. *The Practice of Enterprise Modeling*. España: Springer, Cham.
- Bass, L. (2018). The Software Architect and DevOps. *IEEE Software*, Volumen 35, 8 - 10.
- Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A software architect's perspective*. Addison-Wesley.
- BBVA. (06 de Abril de 2018). *BBVA - El dato de hoy habla de DevOps*. Obtenido de <https://www.bbva.com/es/dato-hoy-habla-devops/>
- BBVA. (31 de Diciembre de 2018). *Memoria 2018*. Obtenido de <https://extranetperu.grupobbva.pe/memoria2018/engineering.html>
- Bejarano De la Hoz, P. A., & Heredia Guerrero, A. M. (2019). Estrategia de DevOps para aportar calidad en los productos de software en MIPYMES de Colombia. Santiago de Cali, Colombia.
- Berta Hinostroza, O. (Noviembre de 2011). Incorporación de la integración continua en el desarrollo de software: caso de estudio: organismo supervisor de la inversión en energía y minería. Piura, Peru.
- Callanan, M., & Spillane, A. (2016). DevOps: Making it easy to do the right thing. *IEEE Software Volumen 33*, 53-59.
- Capodieci, A., Mainetti, L., & Manco, L. (2014). A Case Study to Enable and Monitor Real IT Companies Migrating from Waterfall to Agile. *Proceedings of the 14th international conference on Computational Science*. Guimaraes, Portugal.
- Carrasco, S. (2006). *Metodología de la investigación científica*. Lima: San Marco Editores.
- Carvallho Faustino, J. P. (Setiembre de 2018). DevOps Practices in Incident Management Process. *Dissertation submitted as partial fulfilment of the requirements for the degree of Master in Computer Engineering*. Lisboa.
- Christensen, H. (Julio de 2016). Teaching DevOps and cloud computing using a cognitive apprenticeship and story-telling approach. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE*. New York, USA: Bilkent University.
- Cois, C., Yankel, J., & Connell, A. (13-14 de 10 de 2014). Modern DevOps: Optimizing software. *IEEE International Professional Communication Conference (IPCC)*. Pittsburgh, USA: IEEE.
- Davis, J., & Daniels, R. (2016). *Effective DevOps*. Boston: O'Reilly.
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software Volumen 33*, 94-100.

- Faustino, J. P. (Setiembre de 2018). DevOps Practices in Incident Management Process. *DevOps Practices in Incident Management Process*.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate, The Science behind of DevOps*. Portland: IT Revolution Press.
- Herder, A., Farias, P., & Albuquerque, A. (2016). An Agile Approach to Improve Process-Oriented Software Development. Suiza.
- Hernández, R., Fernández, C., & Batista, P. (2014). *Metodología de la investigación*. Mexico: Mc Graw Hill.
- Hosono, S. (2012). A DevOps framework to shorten delivery time for cloud applications. *International Journal of Computational Science and Engineering - Vol. 7*, 329.
- Hütermann, M. (2012). *Devops for Developers*. Apress.
- Iñiguez Sánchez, L. A. (Junio de 2017). Arquitectura tecnológica para la entrega continua de software con despliegue en contenedores. Cuenca.
- Kerzazi, N., & Adams, B. (2016). Who Needs Release and DevOps Engineers, and Why? *2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED)* (pág. 1). Austin, Texas, USA: IEEE.
- Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2016). *The DevOps handbook*. Portland: IT Revolution Press.
- Lopez Aguilar, F. (7 de Diciembre de 2015). Aprovisionamiento automático de infraestructura y de configuración de la infraestructura. El caso Adblock Plus. Zacatecas.
- Lwakatare, L., & Karvonen, T. (5-8 de 01 de 2016). Towards DevOps in the Embedded Systems Domain: Why is It So Hard? *2016 49th Hawaii International Conference on System Sciences (HICSS)*. Koloa: USA.
- Lwakatare, L., Kuvaja, P., & Oivo, M. (16 de 05 de 2015). Dimensions of DevOps. *Agile Processes in Software Engineering and Extreme Programming. XP 2015*. Suiza: Springer Cham.
- Nebel, A. (Octubre de 2018). Arquitectura de Microservicios para Plataformas de Integración. *Arquitectura de Microservicios para Plataformas de Integración*. Montevideo, Uruguay.
- New Relic. (2015). DevOps without measurment is a fail. New Relic.
- Nybom, K., Smeds, J., & Porres, I. (2016). On the Impact of Mixing Responsibilities Between Devs and Ops. *Agile Processes, in Software Engineering, and Extreme Programming: 17th International Conference, XP 2016* (págs. 131-143). Turku, Finlandia: Springer International Publishing.
- O. White, K. B. M. Manager, and J. KabanovCEO. (2013). *DevOps productivity report 2013*. ZeroTurnaround.

- Perera, P., Silva, R., & Perera, I. (6-9 de 09 de 2017). Improve software quality through practicing DevOps. *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*. Colombo, Sri Lanka: IEEE.
- Puppet. (12 de 09 de 2016). *Puppet, 2016 State of DevOps Report*. Obtenido de <https://puppet.com/resources/report/2016-state-devops-report>
- Rajkumar, M., Pole, A., Adige, V., & Mahanta, P. (8-9 de 4 de 2016). DevOps culture and its impact on cloud delivery and software development. *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*. Dehradun, India: IEEE.
- Retamal Valenzuela, J. H. (2019). Plataformas de desarrollo de aplicaciones en el DCC basada en técnicas DevOps. Santiago de Chile, Chile.
- Smeds, J., Nybom, K., & Porres, I. (05 de 2015). DevOps: A definition and perceived adoption impediments. *Agile Processes in Software Engineering and Extreme Programming* (págs. 166-177). Helsinki, Finlandia: Springer Cham.
- Spinellis, D. (2012). Don't Install Software by Hand. *IEEE Software - Volumen 29*, 86 - 87.
- Suaréz, P. (2011). *Curso de Metodología de la Investigación. Población de estudio y muestra*. Obtenido de http://udocente.sespa.princast.es/documentos/Metodologia_Investigacion/Presentaciones/4_%20poblacion&muestra.pdf
- Tomas, N., Li, J., & Huang, H. (3-4 de 06 de 2019). An Empirical Study on Culture, Automation, Measurement, and Sharing of DevSecOps. *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. Oxford, Inglaterra: IEEE.
- Villavicencio Zambrano, J. F. (2017). Modelo de gestión para el control de cambios y la dirección de proyectos basados en ITIL y Metodologías Ágiles de un departamento de desarrollo de Software. Cuenca, Ecuador.
- Virmani, M. (22 de 04 de 2015). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. *Fifth International Conference on the Innovative Computing Technology (INTECH 2015)*. Pontevedra, España: IEEE.
- Wahaballa, A., Abdellatief, M., Xiong, H., & Qin, Z. (23-25 de 09 de 2015). Toward unified DevOps model. *6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. Beijing, China: IEEE.
- Wettinger, J., Andrikopoulos, V., & Leymann, F. (9-13 de 3 de 2015). Automated capturing and systematic usage of DevOps knowledge for cloud applications. Tempe, USA: IEEE.
- Wettinger, J., Andrikopoulos, V., & Leymann, F. (2015). Enabling DevOps collaboration and continuous delivery using diverse application environments. *OTM 2015*

Conferences. Springer Science + Business Media (págs. 348-358). Rodas, Grecia: Christophe Debruyne.

- Wettinger, J., Breitenbücher, U., & Kopp, O. (2016). Streamlining DevOps automation for cloud applications using TOSCA as standardized metamodel. *Future Generation Computer Systems*, 317 - 332.
- Wettinger, J., Breitenbücher, U., & Leymann, F. (8-11 de 12 de 2014). Standards-Based DevOps Automation and Integration Using TOSCA. *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. Londres, Reino Unido: IEEE.
- Wettinger, J., Breitenbücher, U., & Leymann, F. (2015). Compensation and Convergence - comparing and combining deployment automation approaches. *International Journal of Cooperative Information Systems - Vol 24*, 1.
- Wettinger, J., Breitenbucher, U., & Leymann, F. (02 de 02 de 2015). Standards-based DevOps automation and integration using TOSCA. Londres, Reino Unido: IEEE.
- Woods, E. (2016). Operational: The forgotten architectural view. *IEEE Software*, vol. 33, 20-23.
- Zaid Abrahams, M., & Langerman, J. (24-26 de 09 de 2018). Compliance at Velocity within a DevOps Environment. *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. Berlin, Alemania: IEEE.
- Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and Its Practices. *IEEE Software - Volumen 33*, 32-34.

ANEXOS

Anexo 01 – Matriz de consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	OPERACIONALIZACIÓN DE VARIABLES				METODOLOGÍA
General	General	General	V.I.	Concepto	Dimensiones	Indicadores	
¿Cuál es el efecto de Devops sobre el Proceso de Integración e Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019?	Identificar el efecto de Devops sobre el Proceso de Integración e Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.	Existe efecto de DEVOPS sobre el Proceso de Integración e Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019.	DEVOPS	DevOps es ese conjunto de normas culturales y practicas tecnologicas que habilitan un flujo mas rápido desde un trabajo lanificado hacia desaroollo, pruebas, operaciones; mientras prevalece la seguridad, estabilidad y disponibilidad.(Kim, 2016)	No aplica		Diseño de investigación: Pre-Experimental Tipo de investigación: Aplicada Enfoque: Cuantitativo Método de investigación: Hipotético-Deductivo Población: 8 sprints y 18 historias de usuario registradas en 16 semanas Técnicas: Fichaje Instrumentos: Ficha de observación
Especificos	Especificos	Especificos	V.D.	Concepto			
¿Cuál es el efecto de Devops sobre la velocidad de la Integración e Despliegue de Software?	Identificar el efecto Devops sobre la velocidad de Integración e Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019	Existe efecto de Devops sobre la velocidad del proceso de Integración e Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019	Proceso de Integracion y Despliegue de Software	Es el proceso que consiste en combinar los cambios realizados en el código del proyecto, en un repositorio central de forma periódica, para luego ejecutar pruebas y detectar y reparar errores lo antes posible para luego desplegar el software.(Debois,2015)	Velocidad	Cycletime $CT = DD+DT+DP$ Frecuencia de liberación de código $CD = \sum \text{Despliegue}$	
¿Cuál es el efecto de Devops sobre la calidad de la Integración y Despliegue de Software?	Identificar el efecto de Devops sobre la calidad de la Integración e Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019	Existe efecto de Devops sobre la calidad del proceso de Integración e Despliegue de Software en el Sector de Seguros del distrito de San Isidro, Lima, 2019			Calidad	Ratio de éxito $RE = DE/TD*100\%$	

Anexo 02 - Validez de instrumentos

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE EL PROCESO DE INTEGRACION Y DESPLIEGUE DE SOFTWARE.

VELOCIDAD - Cyletime	Pertinencia 1		Relevancia 2		Claridad 3	
	SI	NO	SI	NO	SI	NO
Sprint	✓		✓		✓	
Historia de Usuario	✓		✓		✓	
Descripción	✓		✓		✓	
Días de Desarrollo	✓		✓		✓	
Días de Testing	✓		✓		✓	
Días de despliegue producción	✓		✓		✓	
Cyletime	✓		✓		✓	

VELOCIDAD - Frecuencia de liberación de código	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Cantidad de despliegue por sprint	✓		✓		✓	

CALIDAD - Ratio de Éxito	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Despliegue Exitosos por Sprint	✓		✓		✓	
Despliegues Fallidos por Sprint	✓		✓		✓	
Total de Despliegues por Sprint	✓		✓		✓	
Ratio de Éxito	✓		✓		✓	

Observaciones (Precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable (X) Aplicable después de corregir () No aplicable ()

Apellidos y nombres del juez validador:

ZEVALLOS DELCADO CILMAR
MASTER EN GESTION ESTRATEGICA

Especialidad del validador:

Lima, 06 de diciembre del 2019.

1Pertinencia: El ítem corresponde al concepto teórico formulado.

2Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.

3Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

011
DNI 29627983

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE EL PROCESO DE INTEGRACION Y DESPLIEGUE DE SOFTWARE.

VELOCIDAD - Cycletime	Pertinencia 1		Relevancia 2		Claridad 3	
	SI	NO	SI	NO	SI	NO
Sprint	✓		✓		✓	
Historia de Usuario	✓		✓		✓	
Descripción	✓		✓		✓	
Días de Desarrollo	✓		✓		✓	
Días de Testing	✓		✓		✓	
Días de despliegue producción	✓		✓		✓	
Cycletime	✓		✓		✓	

VELOCIDAD - Frecuencia de liberación de código	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Cantidad de despliegue por sprint	✓		✓		✓	

CALIDAD - Ratio de Éxito	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Despliegue Exitosos por Sprint	✓		✓		✓	
Despliegues Fallidos por Sprint	✓		✓		✓	
Total de Despliegues por Sprint	✓		✓		✓	
Ratio de Éxito	✓		✓		✓	

Observaciones (Precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable (X) Aplicable después de corregir () No aplicable ()

Apellidos y nombres del juez validador: *Dargan Fobres, Nolasco*

Especialidad del validador: *Inj. Informático - MDA en Administración Estratégica Empresas*

Lima, 06 de diciembre del 2019.

1Pertinencia: El ítem corresponde al concepto teórico formulado.

2Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.

3Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

NF
1013907

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE EL PROCESO DE INTEGRACION Y DESPLIEGUE DE SOFTWARE.

VELOCIDAD - Cycletime	Pertinencia 1		Relevancia 2		Claridad 3	
	SI	NO	SI	NO	SI	NO
Sprint	✓		✓		✓	
Historia de Usuario	✓		✓		✓	
Descripción	✓		✓		✓	
Días de Desarrollo	✓		✓		✓	
Días de Testing	✓		✓		✓	
Días de despliegue producción	✓		✓		✓	
Cycletime	✓		✓		✓	

VELOCIDAD - Frecuencia de liberación de código	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Cantidad de despliegue por sprint	✓		✓		✓	

CALIDAD - Ratio de Éxito	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Despliegue Exitosos por Sprint	✓		✓		✓	
Despliegues Fallidos por Sprint	✓		✓		✓	
Total de Despliegues por Sprint	✓		✓		✓	
Ratio de Éxito	✓		✓		✓	

Observaciones (Precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable (X) No aplicable ()

Apellidos y nombres del juez validador: *Martin Bazo Bustamante*

Especialidad del validador: *ING. DE SISTEMAS - U. Q. M. A.*

Lima, 06 de diciembre del 2019.

RIMAC SEGUROS Y REASEGUROS

MARTIN BAZO BUSTAMANTE
 Gerente Desarrollo TI

1Pertinencia: El ítem corresponde al concepto teórico formulado.

2Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.

3Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO QUE MIDE EL PROCESO DE INTEGRACION Y DESPLIEGUE DE SOFTWARE.

VELOCIDAD - Cyclictime	Pertinencia 1		Relevancia 2		Claridad 3	
	SI	NO	SI	NO	SI	NO
Sprint	✓		✓		✓	
Historia de Usuario	✓		✓		✓	
Descripción	✓		✓		✓	
Días de Desarrollo	✓		✓		✓	
Días de Testing	✓		✓		✓	
Días de despliegue producción	✓		✓		✓	
Cyclictime	✓		✓		✓	

VELOCIDAD - Frecuencia de liberación de código	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Cantidad de despliegue por sprint	✓		✓		✓	

CALIDAD - Ratio de Éxito	Pertinencia		Relevancia		Claridad	
	SI	NO	SI	NO	SI	NO
Ítem	✓		✓		✓	
Sprint	✓		✓		✓	
Despliegue Exitosos por Sprint	✓		✓		✓	
Despliegues Fallidos por Sprint	✓		✓		✓	
Total de Despliegues por Sprint	✓		✓		✓	
Ratio de Éxito	✓		✓		✓	

Observaciones (Precisar si hay suficiencia):

Opinión de aplicabilidad: Aplicable (✓) Aplicable después de corregir () No aplicable ()

Apellidos y nombres del juez validador: *Nancy Salazar Angel*

Especialidad del validador: *Entrenador Certificado del DevOps Institute, Certified DevOps Developer, Certified Continuous Delivery Architect, Bachiller Ingeniería Geomática*

Lima, 06 de diciembre del 2019.

1Pertinencia: El ítem corresponde al concepto teórico formulado.

2Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo.


3Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

[Firma]
43458152

Anexo 03 - Fichas PreTest

FICHA DE OBSERVACIÓN - PRETEST						
Investigador		Fernando Casas Yeren				
Empresa		Rimac Seguros y Reaseguros				
Dirección		Calle Las Begonias 540 - San Isidro				
Proceso Observador		Integración y Despliegue de Software				
INDICADOR	DESCRIPCIÓN	TÉCNICA	UNIDAD DE MEDIDA	INSTRUMENTO	FORMULA	
Cycletime	Se evaluará el tiempo de desarrollo de las historias de usuario desde desarrollo hasta su entrega en producción	Fichaje	Días	Ficha de Observación	CT = DD+DT+DP CT= Cycletime DD= Días de Desarrollo DT= Días de Testing DP= Días de despliegue producción	
SPRINT	HU	DESCRIPCIÓN	DD	DT	DP	CT
14	US1750	Taggeo Adobe Analytics	16	5	1	22
14	US1749	Recibir Suma Asegurada Adhoc	16	5	1	22
15	US1751	Recibir nueva oferta Temporal Total	8	5	1	14
16	US1753	Flujo Lineal Temporal Total	19	2	1	22
16	US1752	Probar Pivot ADN	19	2	1	22
16	US1732	HOTFIX TAG ERRORES Y COOKIE DE ANALYTICS	14	5	0	19
17	US1755	Seguridad en Journey	2	2	0	4
17	US1754	Mejorar la Performance Journey	10	7	1	18
18	US1813	Taggeo Analytics para Pivot y TT	4	3	5	12
18	US1838	Bugs y Nuevas Funcionalidades Pivot Meta	8	0	0	8
19	US1856	Refrescar LOGIN Journey	5	2	1	8
19	US1859	Automatización flujo hipotecario	8	2	0	10
20	US1909	Migrar tablas RedShift - MySQL	1	1	1	3
21	US1761	Meta - Educacion Hijos	9	20	0	29
21	US1758	Perfilamiento Flexivida - Nuevo VFP	6	20	0	26
21	US1757	Cotizar Core Acel E - Nuevo Flexivida - BACK	8	18	3	29
21	US1910	Cotizar Core Acel E - Nuevo Flexivida - FRONT	8	18	3	29
21	US1795	Tabla de Valores de Proyectos Flexivida - Nuevo VFP	9	20	0	29
Total			170	137	19	18.11

RIMAC SEGUROS Y REASEGUROS

MARTIN BAZO BUSTAMANTE
Gerente Desarrollo TI

FICHA DE OBSERVACION - PRETEST					
Investigador			Fernando Casas Yeren		
Empresa			Rímac Seguros y Reaseguros		
Dirección			Calle Las Begonias 540 - San Isidro		
Proceso Observador			Integración y Despliegue de Software		
INDICADOR	DESCRIPCIÓN	TÉCNICA	UNIDAD DE MEDIDA	INSTRUMENTO	FORMULA
Frecuencia de liberación de código	Se evaluará las veces que se libera historias de Usuario a producción por sprint.	Fichaje	Días	Ficha de Observación	CD=Cantidad de Despliegues por Sprint

ITEM	SPRINT	CD
1	Sprint 14	1
2	Sprint 15	1
3	Sprint 16	1
4	Sprint 17	2
5	Sprint 18	2
6	Sprint 19	2
7	Sprint 20	3
8	Sprint 21	1
Total		1.63

RIMAC SEGUROS Y REASEGUROS


MARTÍN BAZO BUSTAMANTE
Gerente Desarrollo TI

FICHA DE OBSERVACION - PRETEST					
Investigador			Fernando Casas Yeren		
Empresa			Rímac Seguros y Reaseguros		
Dirección			Calle Las Begonias 540 - San Isidro		
Proceso Observador			Integración y Despliegue de Software		
INDICADOR	DESCRIPCIÓN	TÉCNICA	UNIDAD DE MEDIDA	INSTRUMENTO	FORMULA
Ratio de éxito	Se evaluará ratio código de desplegado de manera exitosa, no requiere remediación, por Sprint	Fichaje	Porcentaje	Ficha de Observación	$RE = DE/TD * 100\%$ $TD = DE + DF$ DE=Despliegue Exitosos por Sprint DF=Despliegues Fallidos por Sprint TD=Total de Despliegues por Sprint RE= Ratio de Éxito

ITEM	SPRINT	DE	DF	TD	RE
1	Sprint 14	1	1	2	50%
2	Sprint 15	1	0	1	100%
3	Sprint 16	1	0	1	100%
4	Sprint 17	1	1	2	50%
5	Sprint 18	1	1	2	50%
6	Sprint 19	1	4	5	20%
7	Sprint 20	0	1	1	0%
8	Sprint 21	2	1	3	67%
Total		8	9	17	54.58%


RIMAC SEGUROS Y REASEGUROS

MARTIN BAZO BUSTAMANTE
Gerente Desarrollo TI

Anexo 04 - Fichas PosTest

FICHA DE OBSERVACION - POSTEST						
Investigador		Fernando Casas Yeren				
Empresa		Rimac Seguros y Reaseguros				
Direccion		Calle Las Begonias 540 - San Isidro				
Proceso Observador		Integracion y Despliegue de Software				
INDICADOR	DESCRIPCION	TÉCNICA	UNIDAD DE MEDIDA	INSTRUMENTO	FORMULA	
Cycletime	Se evaluará el tiempo de desarrollo de las historias de usuario desde desarrollo hasta su entrega en produccion	Fichaje	Días	Ficha de Observación	CT = DD+DT+DP CT= Cycletime DD= Dias de Desarrollo DT= Dias de Testing DP= Dias de despliegue produccion	

SPRINT	HU	DESCRIPCION	DD	DT	DP	CT
22	US1858	Portafolio Nuevo VFP (Flexivida)	9	8	0	17
22	US1947	Emitir Polizas Acsel-E	15	3	0	18
23	US1948	Crear RI Core	5	6	1	12
24	US1949	Validador de informacion	13	1	3	17
25	US1950	Pagos primera prima	9	1	3	13
25	US1945	Roles en AIM	7	1	1	9
26	US1952	Registro de medios de contacto	6	2	1	9
26	US1953	Autorizacion de LDPDP	6	2	1	9
27	US1954	Perfilador v 1.0	3	1	0	4
27	US1955	Emission en Acsel-E sin crear tercero	8	5	4	17
27	US1960	Perfilador v 2.0	8	1	8	17
28	US1956	Formulario Unico Flexivida	7	1	5	13
28	US1957	Obtener Cumulos	9	4	4	17
28	US1958	Suma Asegurada Adhoc	8	2	6	16
29	US1959	Prima Ahorro VFP	2	1	3	6
29	US1961	Modificar Estilos All Finanz	5	1	1	7
29	US1962	Integracion AllFinanz	1	4	0	5
29	US1951	Recibir Alerta cuando se pierda conexio GPRS	1	2	0	3
Total			122	46	41	11.61

RIMAC SEGUROS Y REASEGUROS

MARTÍN BAZO BUSTAMANTE
 Gerente Desarrollo TI

FICHA DE OBSERVACION - POSTTEST					
Investigador			Fernando Casas Yeren		
Empresa			Rímac Seguros y Reaseguros		
Direccion			Calle Las Begonias 540 - San Isidro		
Proceso Observador			Integracion y Despliegue de Software		
INDICADOR	DESCRIPCIÓN	TÉCNICA	UNIDAD DE MEDIDA	INSTRUMENTO	FORMULA
Frecuencia de liberación de código	Se evaluará las veces que se libera historias de Usuario a produccion por sprint	Fichaje	Días	Ficha de Observación	CD=Cantidad de Despliegues por Sprint

ITEM	SPRINT	CD
1	Sprint 22	1
2	Sprint 23	2
3	Sprint 24	3
4	Sprint 25	5
5	Sprint 26	1
6	Sprint 27	5
7	Sprint 28	5
8	Sprint 29	4
Total		3.25

RIMAC SEGUROS Y REASEGUROS

MARTIN BAZO BUSTAMANTE
Gerente Desarrollo TI

FICHA DE OBSERVACION - POSTEST					
Investigador			Fernando Casas Yeren		
Empresa			Rlmac Seguros y Reaseguros		
Direccion			Calle Las Begonias 540 - San Isidro		
Proceso Observador			Integracion y Despliegue de Software		
INDICADOR	DESCRIPCION	TÉCNICA	UNIDAD DE MEDIDA	INSTRUMENTO	FORMULA
Ratio de exito	Se evaluará ratio codigo de desplegado de manera exitosa, no requiere remediacion, por Sprint	Fichaje	Porcentaje	Ficha de Observación	$RE = DE/TD * 100\%$ $TD = DE + DF$ DE=Despliegue Exitosos por Sprint DF=Despliegues Fallidos por Sprint TD=Total de Despliegues por Sprint RE= Ratio de Exito

ITEM	SPRINT	DE	DF	TD	RE
1	Sprint 22	1	0	1	100%
2	Sprint 23	3	0	3	100%
3	Sprint 24	2	1	3	67%
4	Sprint 25	1	1	2	50%
5	Sprint 26	1	1	2	50%
6	Sprint 27	4	1	5	80%
7	Sprint 28	7	3	10	70%
8	Sprint 29	6	3	9	67%
Total		25	10	35	72.92%

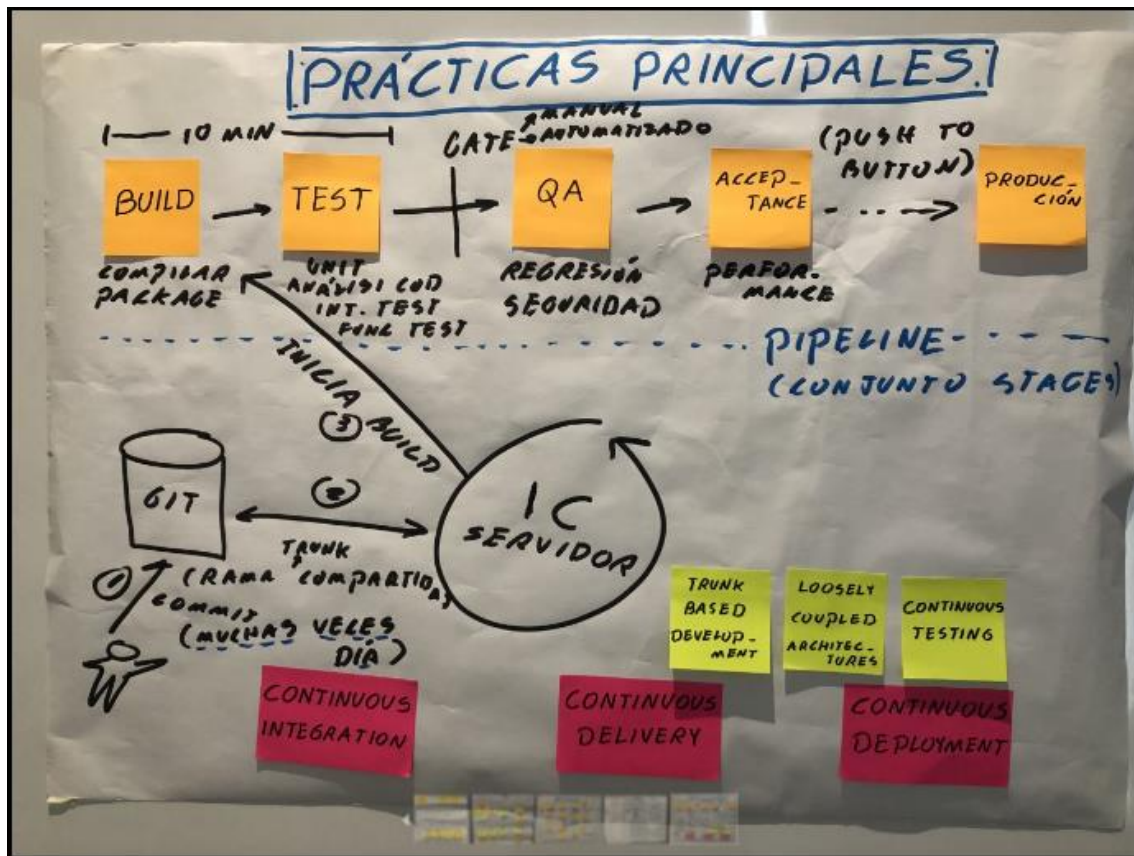
RIMAC SEGUROS Y REASEGUROS

MARTIN BAZO BUSTAMANTE
Gerente Desarrollo TI

1- Current State del proceso de integración y despliegue de software del proyecto Journey Venta Vida.

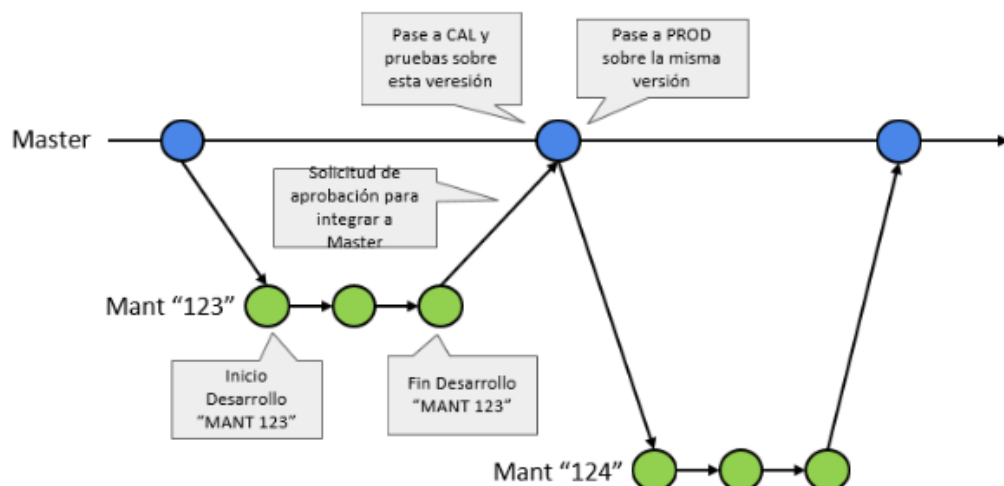


2- Estrategia de DevOps implementada.



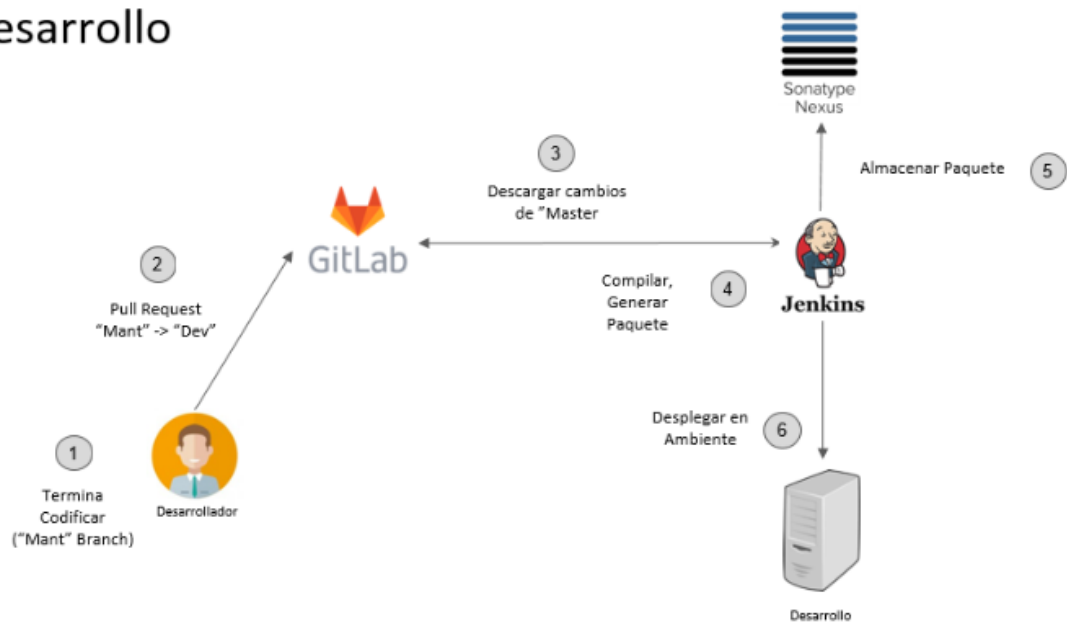
3- Estrategia de Ramas implementadas para la integración continúa.

Flujo Básico: Desarrollo, Calidad y Producción

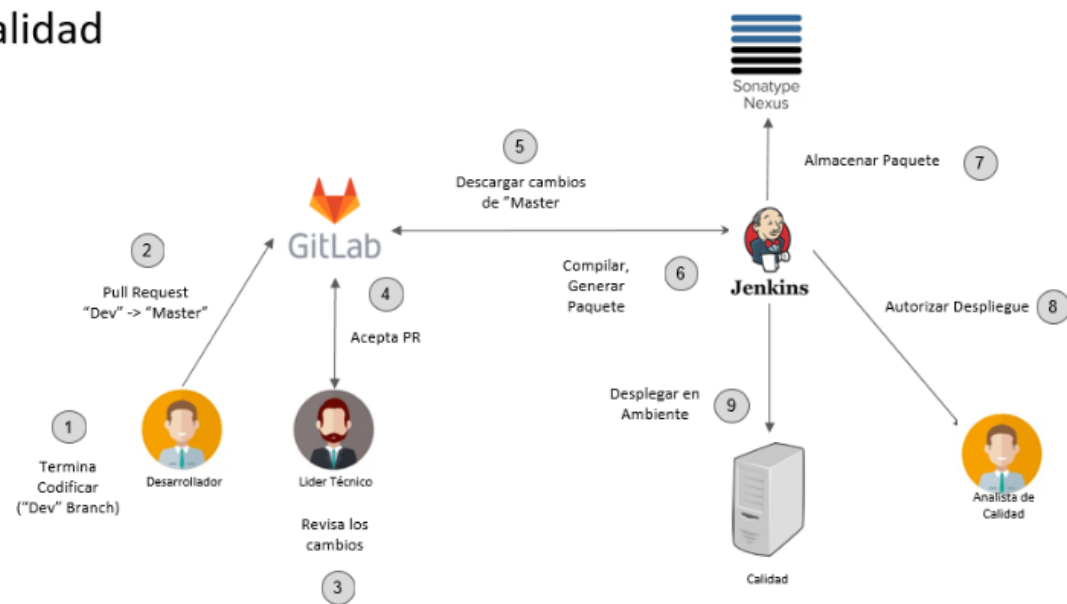


4- Flujos de aprobación y roles para integración y despliegue continuo.

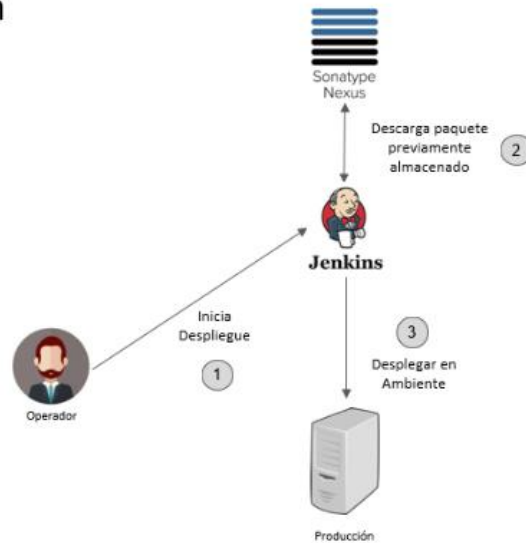
Desarrollo



Calidad



Producción



5- Process Block y Owner Devops

Process Block Owner
Responsable de alcanzar el Next Target de su Process Block.

Process Block Owners							
Desarrollo	Despliegue en Calidad	Pruebas de Calidad	Regresión	Pruebas de Usuario	Pruebas Seguridad	CAB	Despliegue Producción
Owner: Fernando	Owner: Fernando	Owner: Yeiny y Luis P.		Owner: Yeiny	Owner: Giselle y Guillermo	Owner: Walter	Owner: Walter
Oportunidades: <ul style="list-style-type: none"> - Reducir el Batch de trabajo - Traer Testing y Seguridad hacia Dev. - Estrategia de Versionamiento - Integración Continua. 	Oportunidades: <ul style="list-style-type: none"> - Automatizar la ejecución. - Optimizar las aprobaciones y artefactos. 	Oportunidades: <ul style="list-style-type: none"> - Automatizar las pruebas. - Reducir los ciclos entre desarrollo y calidad. - Ejecutar testing de manera temprana desde desarrollo. 		Oportunidades: <ul style="list-style-type: none"> - Reducir tiempo de aprobación. - Mover pruebas de usuario hacia calidad. 	Oportunidades: <ul style="list-style-type: none"> - Automatizar pruebas de seguridad. - Embeber seguridad desde desarrollo. - Security Advisor dedicado. - Mejorar lineamientos y capacitar en ellos. 	Oportunidades: <ul style="list-style-type: none"> - Mejorar los tiempos de aprobación velando por la estabilidad. - Diversas Políticas de CAB según el riesgo. - Aprobaciones Automatizadas para ciertos riesgos. 	Oportunidades: <ul style="list-style-type: none"> - Automatizar la ejecución. - Persona Dedicada de Operaciones.
Oportunidades: <ul style="list-style-type: none"> - Pipeline de Continuous Delivery y Release - Métricas de Tiempo, Calidad y Estabilidad 							

RIMAC

6- Historias de Usuario y Sprints del proyecto Journey Venta Vida - Aplicación Central Agile.

ca Journey Venta Vida Plan Track Quality Portfolio Reports						
Sprint 13	30/07 - 06/08				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1749	Recibir Suma Asegurada Ad-hoc					
Sprint 14	06/08 - 13/08				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1750	Taggeo Adobe Analytics					
Sprint 15	13/08 - 20/08				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1751	Recibir nueva oferta Temporal Total					
Sprint 16	20/08 - 27/08				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1753	Flujo Lineal Temporal Total					
US1752	Probar Pivot ADN					
US1732	HOTFIX PASE A PROD - AGREGAR TAG ERRORES Y COOKIE PARA A...					
Sprint 17	27/08 - 03/09				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1755	Seguridad en Journey					
US1754	Mejorar la Performance Journey					
Sprint 18	03/09 - 10/09				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1813	Taggeo Analytics para Pivot y TT					
Sprint 18	03/09 - 10/09				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1813	Taggeo Analytics para Pivot y TT					
US1838	Bugs y Nuevas Funcionalidades Pivot Meta					
Sprint 19	10/09 - 17/09				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1856	Refrescar LOGIN Journey					
US1857	Id Cliente Único					
US1859	Automatización flujo hipotecario					
Sprint 20	17/09 - 24/09				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1909	Migración Redshift a MySQL					
Sprint 21	24/09 - 01/10				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	
US1761	Meta Educación Hijos					
US1758	Perfilamiento Flexivida - Nuevo VFP					
US1757	Cotizar Core Accel E - Nuevo Flexivida (Back)					
US1910	Cotizar Core Accel E - Nuevo Flexivida (Front)					
US1795	Tabla de Valores de Proyectos Flexivida - Nuevo VFP					
US1858	Portafolio Nuevo VFP (Flexivida)					
US1947	Emitir Póliza en el Accel-E					
Sprint 22	01/10 - 08/10				0%	
ID	NAME	DEPENDENCIES	SCHEDULE STATE	TASKS	EST	

7- Integración continua del proyecto Journey Venta Vida– Gitlab

The screenshot displays the GitLab web interface for the project 'journey-vida-web' (Project ID: 216). The left sidebar shows the project navigation menu with options like Project, Repository, Issues, Merge Requests, CI / CD, Operations, Wiki, Snippets, and Settings. The main content area shows the project details, including a table of files and a list of merge requests.

Project Details:

- Project ID: 216
- 1,656 Commits
- 28 Branches
- 231 Tags
- 5.8 MB Files
- Repository description: Repositorio del proyecto front de Journey ventas vida.
- Current branch: master
- Recent merge: Merge branch 'hotfix' into 'master' by Miguel Rodriguez 1 hour ago.
- Buttons: README, Add CHANGELOG, Add CONTRIBUTING, Add Kubernetes cluster, CI/CD configuration.

Files Table:

Name	Last commit	Last update
public	fix(bugs) add solution to translate, work and cover...	2 months ago
src	fix(url) change modal to urlhttps	1 hour ago
.editorconfig	Initial commit. Migration from old repository	3 months ago
.gitignore	feat(flexivida) get product in form	2 months ago
.gitlab-ci.yml	Initial commit. Migration from old repository	3 months ago
.secrets.baseline	add baseline for detect secrets	6 days ago
Jenkinsfile	Update Jenkinsfile	6 days ago

Merge Requests Table:

Branch	Author	Status	Updated
product	1300 - opened 1 day ago by Marco Solimano	MERGED	updated 1 day ago
Feat/bonus	1299 - opened 4 days ago by Jhonatan Rojas	MERGED	updated 4 days ago
Fix/product	1298 - opened 4 days ago by Marco Solimano	MERGED	updated 4 days ago
feat(whatchListx)	1297 - opened 4 days ago by Marco Solimano	MERGED	updated 4 days ago
Fix/product	1296 - opened 4 days ago by Marco Solimano	MERGED	updated 4 days ago
feat(whatchListx)	1295 - opened 4 days ago by Marco Solimano	MERGED	updated 4 days ago
Fix/product	1294 - opened 4 days ago by Marco Solimano	MERGED	updated 4 days ago
feat(whatchList)	1293 - opened 4 days ago by Marco Solimano	MERGED	updated 4 days ago
Hotfix	1292 - opened 4 days ago by Jhonatan Rojas	MERGED	updated 4 days ago
fix(payment) add currency	1291 - opened 5 days ago by Jhonatan Rojas	MERGED	updated 5 days ago
feat(whatchList)	1290 - opened 5 days ago by Marco Solimano	MERGED	updated 5 days ago
Fix/product	1289 - opened 5 days ago by Marco Solimano	MERGED	updated 5 days ago
feat(integracion/cumulos)	1288 - opened 5 days ago by Marco Solimano	CLOSED	updated 5 days ago

8- Despliegue Continuo del proyecto Journey Venta Vida – Jenkins.

The image shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and a 'Pipelines' tab. Below the navigation bar is a 'Cuadro de mando' (Dashboard) section with a 'Nuevo Pipeline' button. The main content area displays a list of pipelines with columns for 'NOMBRE', 'SALUD' (Health), 'RAMAS' (Branches), and 'PETICIÓN DI' (Daily Request).

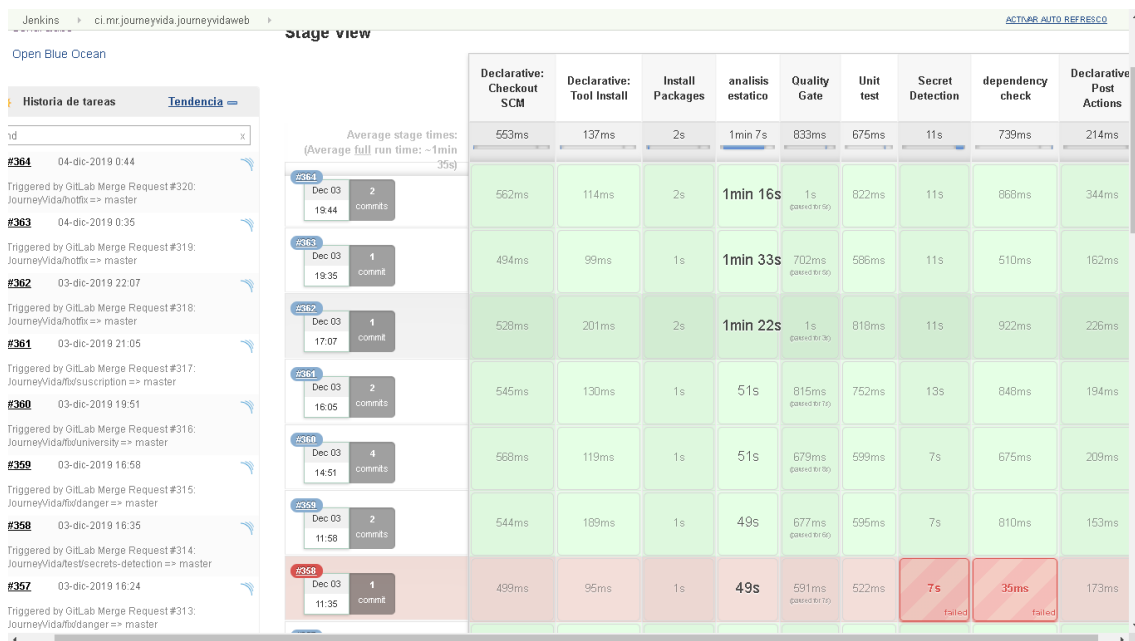
NOMBRE	SALUD	RAMAS	PETICIÓN DI
amr.testdirect.journeyvidaweb		-	-
ci.amr.journeyvida.adn-domain		-	-
ci.amr.journeyvida.common-domain		-	-
ci.amr.journeyvida.cotizacion-domain		-	-
ci.amr.journeyvida.dps-domain		-	-
ci.amr.journeyvida.emision-domain		-	-
ci.amr.journeyvida.journeyvidaweb		-	-
ci.amr.journeyvida.prospeccion-domain		-	-
ci.amr.journeyvida.usuarios-domain		-	-
ci.amr.journeyworkspace.journeyworksiteweb		-	-
ci.amr.journeyworkspace.notificacion-domain		-	-
ci.amr.nineline.journeyvidaweb		-	-

Below the pipeline list, a specific pipeline run is shown for 'ci.amr.journeyvida.journeyvidaweb'. The run details include a 'Commit' section with a green bar indicating the commit was triggered by a GitLab Merge Request. The 'Description' section shows the pipeline was triggered by a GitLab Merge Request. The 'Build Steps' section displays a sequence of steps: Start, Install Packages, análisis estatico, Quality Gate, Unit test, Secret Detection, dependency check, and End. The 'Install Packages' step is highlighted, showing a list of tasks and their durations.

Install Packages - 2s

- ✓ > Check out from version control <1s
- ✓ > Use a tool from a predefined Tool Installation <1s
- ✓ > Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. <1s
- ✓ > Use a tool from a predefined Tool Installation <1s
- ✓ > Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. <1s
- ✓ > Install packages — Print Message <1s
- ✓ > yarn install — Shell Script 1s

✓	180	–	Triggered by GitLab Merge Request #149: JourneyVida/feat/emision => master	4m 41s	a month ago
✓	179	–	Triggered by GitLab Merge Request #148: JourneyVida/feat/emision => master	4m 26s	a month ago
✓	178	–	Triggered by GitLab Merge Request #146: JourneyVida/feat/emision => master	5m 7s	a month ago
✓	177	–	Triggered by GitLab Merge Request #145: JourneyVida/fix/producto => master	5m 8s	a month ago
✓	176	–	Triggered by GitLab Merge Request #144: JourneyVida/feat/emision => master	4m 44s	a month ago
✓	175	–	Triggered by GitLab Merge Request #143: JourneyVida/feat/emision-cotizacion =>	4m 55s	a month ago
✓	174	–	Triggered by GitLab Merge Request #142: JourneyVida/feat/emision-cotizacion =>	4m 46s	a month ago
✓	173	–	Triggered by GitLab Merge Request #141: JourneyVida/feat/emision => master	4m 53s	a month ago
✓	172	–	Triggered by GitLab Merge Request #140: JourneyVida/fix/pdf-filedvda => master	4m 29s	a month ago
✓	171	–	Triggered by GitLab Merge Request #139: JourneyVida/feat/emision => master	4m 30s	a month ago
✓	170	–	Triggered by GitLab Merge Request #138: JourneyVida/feat/emision => master	4m 20s	a month ago
✓	169	–	Triggered by GitLab Merge Request #137: JourneyVida/feat/adobe-analytics => m	4m 26s	a month ago
✓	168	–	Triggered by GitLab Merge Request #136: JourneyVida/feat/adobe-analytics => m	4m 55s	a month ago
✗	167	–	Triggered by GitLab Merge Request #135: JourneyVida/feat/emision => master	4m 27s	a month ago
✗	166	–	Triggered by GitLab Merge Request #134: JourneyVida/fix/console => master	4m 27s	a month ago
✗	165	–	Triggered by GitLab Merge Request #133: JourneyVida/fix/console => master	4m 42s	a month ago



Anexo 06 - Carta de presentación



UNIVERSIDAD CÉSAR VALLEJO

Escuela de Posgrado

"Año de la lucha contra la corrupción y la impunidad"

Lima, 13 de diciembre de 2019

Carta P. 562-2019-EPG-UCV-LN

ING. MARTIN BAZO BUSTAMANTE
GERENTE DE DESARROLLO DE TI
VICE PRESIDENCIA DE TECNOLOGIA DE LA INFORMACIÓN
RIMAC SEGUROS Y REASEGUROS

De mi mayor consideración:

Es grato dirigirme a usted, para presentar a **FERNANDO ENRIQUE CASAS YEREN** identificado con DNI N.° 40973608 y código de matrícula N.° 7001221485; estudiante del Programa de **MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN** quien se encuentra desarrollando el Trabajo de Investigación (Tesis):

IMPLEMENTACIÓN DE DEVOPS PARA MEJORAR LA INTEGRACIÓN Y DESPLIEGUE DE SOFTWARE EN EL SECTOR DE SEGUROS

En ese sentido, solicito a su digna persona otorgar el permiso y brindar las facilidades a nuestro estudiante, a fin de que pueda desarrollar su trabajo de investigación en la institución que usted representa. Los resultados de la presente serán alcanzados a su despacho, luego de finalizar la misma.

Con este motivo, le saluda atentamente,



Dr. Carlos Ventura Orbegoso
Jefe de la Escuela de Posgrado
Universidad César Vallejo - Campus Lima Norte

RCQA

Somos la universidad de los
que quieren salir adelante.



ucv.edu.pe

Anexo 07 - Carta de aceptación



“Año de la Lucha Contra la Corrupción y la Impunidad”

Lima, 06 de diciembre de 2019

CARTA DE ACEPTACIÓN

De nuestra consideración:

Por el presente, en mi calidad de Gerente Desarrollo TI de la empresa Rimac Seguros y Reaseguros con RUC 20100041953, autorizo al Ingeniero Fernando Enrique Casas Yeren, estudiante de la Maestría en Ingeniería de Sistemas con mención en Tecnologías de la Información de la Universidad Cesar Vallejo, identificado con DNI 40973608, a recolectar información sobre la Implementación de DEVOPS y del Proceso de Integración y Despliegue de la empresa que represento, con el propósito de culminar el estudio de su tesis para el proceso en mención.

Sin otro particular me despido.

Atentamente,

RIMAC SEGUROS Y REASEGUROS
.....
MARTIN BAZO BUSTAMANTE
Gerente Desarrollo TI

ACTA DE APROBACIÓN DE ORIGINALIDAD DE TRABAJO ACADÉMICO

Yo, Angel Salvatierra Melgar, docente de la Escuela de Posgrado de la Universidad César Vallejo filial Lima Norte. La tesis titulada **"Implementación de devops para mejorar la integración y despliegue de software en el sector de seguros"** de la estudiante **Fernando Enrique Casas Yeren**, constato que la investigación tiene un índice de similitud de 9% verificable en el reporte de originalidad del programa Turnitin.

La suscrita analizó dicho reporte y concluyo que cada una de las coincidencias detectadas no constituye plagio. A mi leal saber y entender la tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

Lima, 08 de enero del 2020



Angel Salvatierra Melgar

DNI:19873533



UNIVERSIDAD CÉSAR VALLEJO

Centro de Recursos para el Aprendizaje y la Investigación (CRAI)
"César Acuña Peralta"

FORMULARIO DE AUTORIZACIÓN PARA LA PUBLICACIÓN ELECTRÓNICA DE LAS TESIS

1. DATOS PERSONALES

Apellidos y Nombres: (solo los datos del que autoriza)

CASAS YEREN FERNANDO ENRIQUE

D.N.I. : 40973608

Domicilio : Pss. Sociología 171-Los Olivos

Teléfono : Fijo : 605-1076 Móvil : 996855657

E-mail : FERCASAS81@GMAIL.COM

2. IDENTIFICACIÓN DE LA TESIS

Modalidad:

☐ Tesis de Pregrado

Facultad :

Escuela :

Carrera :

Título :

☒ Tesis de Posgrado

☒ Maestría

☐ Doctorado

Grado : MAESTRO EN INGENIERÍA DE SISTEMAS

Mención : TECNOLOGÍAS DE LA INFORMACIÓN

3. DATOS DE LA TESIS

Autor (es) Apellidos y Nombres:

CASAS YEREN FERNANDO ENRIQUE

Título de la tesis:

IMPLEMENTACIÓN DE DEVOPS PARA MEJORAR
LA INTEGRACIÓN Y DESPLIEGUE DE SOFTWARE
EN EL SECTOR DE SEGUROS.

Año de publicación : 2020

4. AUTORIZACIÓN DE PUBLICACIÓN DE LA TESIS EN VERSIÓN ELECTRÓNICA:

A través del presente documento, autorizo a la Biblioteca UCV-Lima Norte, a
publicar en texto completo mi tesis.

Firma : 

Fecha : 11-02-2020



UNIVERSIDAD CÉSAR VALLEJO

AUTORIZACIÓN DE LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN

CONSTE POR EL PRESENTE EL VISTO BUENO QUE OTORGA EL ENCARGADO DE INVESTIGACIÓN DE

ESCUELA DE POSGRADO

A LA VERSIÓN FINAL DEL TRABAJO DE INVESTIGACIÓN QUE PRESENTA:

CASAS PEREN FERNANDO ENRIQUE

INFORME TITULADO:

IMPLEMENTACIÓN DE DEVOPS PARA MEJORAR
LA INTEGRACIÓN Y DESPLIEGUE DE SOFTWARE
EN EL SECTOR DE SEGUROS

PARA OBTENER EL TÍTULO O GRADO DE:

MAESTRO EN INGENIERÍA DE SISTEMAS
CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

SUSTENTADO EN FECHA: 19 DE ENERO DEL 2020

NOTA O MENCIÓN: APROBADO POR UNANIMIDAD.



[Firma]
FIRMA DEL ENCARGADO DE INVESTIGACIÓN